UCAB Universidad Católica ANDRÉS BELLO

**FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA INFORMÁTICA**

# Sistema web para la administración del servicio de salud odontológica

**Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: 18**

J U R A D O  E X A M I N A D O R

| | | |
|---|---|---|
| Firma: | Firma: | Firma: |
| Nombre: Susana García | Nombre: Gloria Tarrío | Nombre: Rafael Lara |

| | |
|---|---|
| **REALIZADO POR** | **Jorge Arbelo** |
| **PROFESOR GUÍA** | **Ing. Rafael Lara** |
| **FECHA** | **Octubre, 2016** |

UCAB Universidad Católica
ANDRÉS BELLO
Facultad de Ingeniería
Escuela de Ingeniería Informática

# Sistema web para la administración del servicio de salud odontológica

TRABAJO INSTRUMENTAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

como parte de los requisitos para optar al título

de INGENIERO EN INFORMÁTICA

REALIZADO POR **Jorge Arbelo**

TUTOR EMPRESARIAL **Ing. Michelangelo Fazzino**

TUTOR ACADEMICO **Ing. Rafael Lara**

FECHA **Octubre, 2016**

# INDICE DE CONTENIDO

INDICE DE CONTENIDO ..... i
INDICE DE FIGURAS ..... vii
INDICE DE TABLAS ..... viii
INDICE DE APÉNDICE ..... ix
DEDICATORIA ..... xi
SINOPSIS ..... xii
INTRODUCCIÓN ..... 1
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA ..... 5
Necesidades de la empresa ..... 5
I.1 Declaración de Oportunidad ..... 5
I.2 Solución Propuesta ..... 6
I.3 Objetivo General ..... 9
I.4 Objetivos Específicos ..... 9
I.5 Aporte Funcional ..... 9
I.6 Aporte Tecnológico ..... 10
I.7 Alcance ..... 10
I.8 Justificación ..... 18
CAPITULO II: MARCO TEÓRICO ..... 19
II.1. Tratamiento ..... 19
II.2. Control de versiones ..... 19
II.3. IDE (Entorno de Desarrollo Integrado) ..... 19
II.3.1. Microsoft Visual Studio ..... 19
II.3.2. SharpDevelop ..... 20
II.4. Base de Datos ..... 20
II.4.1. Microsoft SQL Server ..... 20
II.4.2. Navicat ..... 21
II.5. Framework ..... 21
II.5.1. Microsoft .NET Framework ..... 21
II.5.2. ASP .NET ..... 22
II.5.3. ASP .NET MVC Framework ..... 22
II.5.4. Entity Framework ..... 22
II.5.5. Twitter Bootstrap ..... 22
II.6. Lenguajes de Programación y de Consulta ..... 23

II.6.1. C# ..... 23
II.6.2. SQL ..... 23
II.7. Servicio Web ..... 24
II.7.1. SOAP ..... 24
II.7.2. REST ..... 25
CAPÍTULO III: MARCO METODOLÓGICO ..... 26
III.1 Metodología RAD ..... 26
III.2 Fases de la Metodología RAD ..... 26
III.3 Características de la Metodología RAD ..... 28
III.4 Justificación ..... 28
CAPÍTULO IV: DESARROLLO ..... 30
IV.1 Investigación Preliminar ..... 30
IV.1.1 Estudio de herramientas tecnológicas ..... 30
IV.1.2 Evaluación y selección de la herramienta de desarrollo ..... 30
IV.1.3 Evaluación y selección de la herramienta de desarrollo de la base de datos ..... 32
IV.1.4 Evaluación y selección del Framework para el Desarrollo Web ..... 35
IV.2 Análisis, Planificación de Requerimientos y Diseño ..... 38
IV2.1. Análisis y Planificación de requerimientos ..... 38
IV.2.1 Diseño ..... 39
IV 2.2.1 Rediseñar Proceso de Gestión de Pagos ..... 39
IV 2.2.2 Rediseñar Proceso de Control de Tratamientos ..... 40
IV.3 Desarrollo ..... 41
IV.3.1 Iteración 1: Diseñar e implementar la base de datos en donde se almacenarán los datos del sistema. ..... 42
IV.3.1.1 Análisis y planificación de requerimientos ..... 42
IV.3.1.2 Diseño ..... 42
IV.3.1.3 Construcción ..... 43
IV.3.1.4 Implementación ..... 45
IV.3.2 Iteración 2: Diseñar e implementar módulo de pacientes ..... 46
IV.3.2.1 Análisis y planificación de requerimientos ..... 46
IV.3.2.2 Diseño ..... 46
IV.3.2.3 Construcción ..... 47

IV.3.2.4 Implementación ..... 48
IV.3.3 Iteración 3: Diseñar e implementar módulo de doctores. ..... 49
IV.3.3.1 Análisis y planificación de requerimientos ..... 49
IV.3.3.2 Diseño ..... 49
IV.3.3.3 Construcción ..... 50
IV.3.3.4 Implementación ..... 50
IV.3.4 Iteración 4: Diseñar e implementar módulo de tratamientos. ..... 52
IV.3.4.1 Análisis y planificación de requerimientos ..... 52
IV.3.4.2 Diseño ..... 52
IV.3.4.3 Construcción ..... 53
IV.3.4.4 Implementación ..... 53
IV.3.5 Iteración 5: Diseñar e implementar módulo de enfermedades ..... 54
IV.3.5.1 Análisis y planificación de requerimientos ..... 54
IV.3.5.2 Diseño ..... 54
IV.3.5.3 Construcción ..... 55
IV.3.5.4 Implementación ..... 55
IV.3.6 Iteración 6: Diseñar e implementar módulo de contratos ..... 56
IV.3.6.1 Análisis y planificación de requerimientos ..... 56
IV.3.6.2 Diseño ..... 56
IV.3.6.3 Construcción ..... 56
IV.3.6.4 Implementación ..... 57
IV.3.7 Iteración 7: Diseñar e implementar la Interfaz de Integración y Comunicación con las Aseguradoras. ..... 58
IV.3.7.1 Análisis y planificación de requerimientos ..... 58
IV.3.7.2 Diseño ..... 58
IV.3.7.3 Construcción ..... 59
IV.3.7.4 Implementación ..... 59

IV.3.8 Iteración 8: Diseñar e implementar módulo de pagos.....60

IV.3.8.1 Análisis y planificación de requerimientos.....60

IV.3.8.2 Diseño.....60

IV.3.8.3 Construcción.....61

IV.3.8.4 Implementación.....61

IV.3.9 Iteración 9: Diseñar e implementar módulo de aseguradoras. .....62

IV.3.9.1 Análisis y planificación de requerimientos.....62

IV.3.9.2 Diseño.....62

IV.3.9.3 Construcción.....62

IV.3.9.4 Implementación.....63

IV.3.10 Iteración 10: Diseñar e implementar módulo de citas.....64

IV.3.10.1 Análisis y planificación de requerimientos.....64

IV.3.10.2 Diseño.....64

IV.3.10.3 Construcción.....65

IV.3.10.4 Implementación.....65

IV.3.11 Iteración 11: Diseñar e implementar módulo de reportes. .....66

IV.3.11.1 Análisis y planificación de requerimientos.....66

IV.3.11.2 Diseño.....66

IV.3.11.3 Construcción.....67

IV.3.11.4 Implementación.....67

IV.3.12 Iteración 12: Módulo de Usuarios y Roles (Administración).....68

IV.3.12.1 Análisis y planificación de requerimientos.....68

IV.3.12.2 Diseño.....68

IV.3.12.3 Construcción.....68

IV.3.12.4 Implementación.....68

CAPÍTULO V: RESULTADOS.....70

Aporte Funcional ..... 70
• Rediseñar Proceso de Gestión de Pagos ..... 70
• Rediseñar Proceso de Control de Tratamientos ..... 71
Aporte Tecnológico ..... 72
CAPITULO VI: CONCLUSIONES Y RECOMENDACIONES ..... 79
VI.1 CONCLUSIONES ..... 79
VI.2 RECOMENDACIONES ..... 80
BIBLIOGRAFÍA ..... 81
GLOSARIO ..... 84
APÉNDICE ..... 85

# INDICE DE FIGURAS

Figura 1: Fases de la metodología RAD ..... 27
Figura 2: Caso de Uso Módulo Pacientes ..... 85
Figura 3: Caso de Uso Módulo Doctores ..... 85
Figura 4: Caso de Uso Módulo Enfermedades ..... 86
Figura 5: Caso de Uso Módulo Contratos ..... 86
Figura 6: Caso de Uso Módulo Citas ..... 87
Figura 7: Caso de Uso Módulo Tratamientos ..... 87
Figura 8: Caso de Uso Módulo Aseguradoras ..... 88
Figura 9: Caso de Uso Módulo Pagos ..... 88
Figura 10: Modelo ER ..... 89

# INDICE DE TABLAS

Tabla 1: Iteraciones. Elaboración Propia ..... 29
Tabla 2: Evaluación de Herramienta de Desarrollo ..... 31
Tabla 3: Evaluación de Herramienta de Desarrollo de Base de Datos. ..... 33
Tabla 4: Evaluación de Framework para Desarrollo Web. ..... 35
Tabla 5: Lista de requerimientos del proyecto. ..... 38
Tabla 6: Listado de actores planteados en el sistema ..... 39

## INDICE DE APÉNDICE

APÉNDICE A. DIAGRAMAS DE CASOS DE USO. ..... 85
APÉNDICE A.1. DIAGRAMA DE CASO DE USO MODULO PACIENTES. ..... 85
APÉNDICE A.2. DIAGRAMA DE CASO DE USO MODULO DOCTORES. ..... 85
APÉNDICE A.3. DIAGRAMA DE CASO DE USO MODULO ENFERMEDADES. ..... 86
APÉNDICE A.4. DIAGRAMA DE CASO DE USO MODULO CONTRATOS. ..... 86
APÉNDICE A.5. DIAGRAMA DE CASO DE USO MODULO CITAS. ..... 87
APÉNDICE A.6. DIAGRAMA DE CASO DE USO MODULO TRATAMIENTOS. ..... 87
APÉNDICE A.7. DIAGRAMA DE CASO DE U5O MODULO ASEGURADORAS. ..... 88
APÉNDICE A.8. DIAGRAMA DE CASO DE USO MODULO PAGOS. ..... 88
APÉNDICE B. MODELO ENTIDAD RELACIÓN. ..... 89
APÉNDICE C. INICIO DE SESIÓN. ..... 90
APÉNDICE D. PANTALLA DE INICIO. ..... 90
APÉNDICE E. MODULO PACIENTES. ..... 91
APÉNDICE E.1. CONSULTAR PACIENTE. ..... 91
APÉNDICE E.2. BORRAR PACIENTE. ..... 91
APÉNDICE F. MODULO DOCTORES. ..... 92
APÉNDICE F.1. CREAR DOCTOR. ..... 92
APÉNDICE F.2. CONSULTAR DOCTOR. ..... 93
APÉNDICE F.3. MODIFICAR DOCTOR. ..... 93
APÉNDICE F.4. BORRAR DOCTOR. ..... 93
APÉNDICE G. MODULO CONTRATOS. ..... 94
APÉNDICE G.1. CREAR CONTRATO. ..... 94
APÉNDICE G.2. MODIFICAR CONTRATO. ..... 94
APÉNDICE H. MODULO CITAS. ..... 95
APÉNDICE H.1. CREAR CITA. ..... 95
APÉNDICE H.2. MODIFICAR CITA. ..... 95
APÉNDICE H.3. BORRAR CITA. ..... 96
APÉNDICE H.4. DETALLE CITA. ..... 96
APÉNDICE H.5. SELECCIÓN. ..... 97
APÉNDICE I. MODULO ENFERMEDADES. ..... 97
APÉNDICE I.1. CREAR ENFERMEDAD. ..... 98
APÉNDICE I.2. MODIFICAR ENFERMEDAD. ..... 98
APÉNDICE I.3. BORRAR ENFERMEDAD. ..... 98
APÉNDICE J. MODULO TRATAMIENTOS. ..... 99
APÉNDICE J.1. CREAR TRATAMIENTO. ..... 99
APÉNDICE J.2. MODIFICAR TRATAMIENTO. ..... 100
APÉNDICE J.3. BORRAR TRATAMIENTO. ..... 100
APÉNDICE K. MODULO ASEGURADORAS. ..... 100
APÉNDICE K.1. CREAR ASEGURADORA. ..... 101
APÉNDICE K.2. MODIFICAR ASEGURADORA. ..... 101
APÉNDICE K.3. BORRAR ASEGURADORA. ..... 101
APÉNDICE L. MODULO PAGOS. ..... 102
APÉNDICE L.1 TIPO DE PAGO. ..... 102
APÉNDICE M. MODULO REPORTES. ..... 103
APÉNDICE N. GRÁFICO DENTAL. ..... 103
APÉNDICE O. MODULO DE USUARIOS Y ROLES (ADMINISTRACIÓN). ..... 104

**APÉNDICE O.1. CREAR USUARIO** ..... 104
**APÉNDICE O.2. ROL** ..... 104

## DEDICATORIA

Primero que todo dar gracias a Dios por ser mi guía y hacer realidad este momento…

Este Trabajo de Grado se lo dedico a mi madre Luz Ruby, una luchadora incansable, digna de admiración, sin ella nada de esto podría ser posible. Siempre estuvo ahí para apoyarme, darme aliento en los momentos en los que más necesitaba su consejo y siempre sin dejarme desfallecer, tengo tanto gracias a ella que espero poder retribuirle algún día todo esto. Eres es el primer motor fundamental en mi vida.

Mención especial a mi abuela Marina que me acompaña desde el cielo hace ya 19 años, al igual que mi madre luchadora incansable a lo largo de toda su vida. Me cuidó y educó, estuvo siempre pendiente de mí hasta el último de sus días, me ayudó a ser mejor persona y me dio su amor en todo momento. Abuelita desde donde te encuentres sé que estas orgullosa de mí. Fuiste, eres y serás mi segundo gran motor.

A mis hermanos, Yesid y Román por estar conmigo animándome en todo momento y a Enrique quien ha sido como un padre para mí, brindando su apoyo y su desinteresado consejo. Debo agradecer también a mi tía Cecilia, mi padrino Eliseo Serrano y a sus hijos, mis primos. En ustedes siempre he encontrado un hogar con las puertas abiertas. Y en general a toda mi familia porque de una u otra forma siempre cuento con su apoyo.

A mi novia Alejandra Villegas porque desde el primer día me ha impulsado a cumplir mis metas y mis sueños, siempre con una palabra de apoyo para motivarme, acompañándome y preocupándose por mí en todo momento. Agradezco a la vida por ponerte en mi camino, sin lugar a duda eres el tercer motor en mi vida.

Finalmente, a mí amigo Luis Quintero, quien más que un amigo ha sido como un hermano, mi compañero en los buenos y no tan buenos momentos en la Universidad.

## SINOPSIS

El presente Trabajo Instrumental de Grado (TIG) intitulado "Sistema web para la administración del servicio de salud odontológica.", describe el desarrollo de un sistema constituido por un aplicativo Web que permite a las instituciones proveedoras del Área Salud Odontológica gestionar distintos trámites en forma virtual y acceder a información de utilidad, y que permite a las instituciones odontológicas y a las aseguradoras tener una completa y correcta gestión de sus recursos utilizados por los pacientes, que le permita disminuir los tiempos de respuesta y garantizar que se cumplan todas las condiciones del servicio con todos sus pacientes y doctores asociados. Así mismo, permite a los analistas de las aseguradoras gestionar el pago de los servicios prestados por los odontólogos de una manera sencilla y eficiente.

El desarrollo del sistema fue llevado a cabo con una metodología ágil denominada RAD (*Rapid Application Development*). Su elección se debe a que tanto sus características como sus fases la hacen ideal para la ejecución del presente TIG. La metodología se basa en la división del proyecto en módulos para ser entregados a corto plazo y en iteraciones no mayores a tres semanas, con lo que se logra que el sistema vaya creciendo iterativamente con el desarrollo y entrega de cada módulo. Este sistema se compone principalmente por 11 módulos con los que se cumple con los objetivos planteados del sistema, cada uno de estos módulos detallado y descrito a lo largo de este documento.

Además, se deja implementada una arquitectura tecnológica que permite el crecimiento del sistema, no sólo se limita a los módulos desarrollados, sino que puede seguir creciendo, añadiendo nuevos módulos que puedan complementarlo.

# INTRODUCCIÓN

El presente trabajo está estructurado y descrito en seis capítulos: planteamiento del problema, marco teórico, marco metodológico, desarrollo, resultados y un capítulo final de conclusiones y recomendaciones. Con cada uno de estos capítulos se espera llevar al lector paso a paso al entendimiento de los objetivos y funcionamiento del sistema.

La forma en que un doctor lleva el historial clínico de sus pacientes puede ayudarlo a ofrecer una mejor atención a los mismos, es por esto que se identificó la oportunidad de ofrecer un sistema odontológico que ayudará a los doctores y a sus asistentes a llevar el control sistematizado de sus historias.

El sistema le permitirá al doctor ver de forma inmediata y en un gráfico toda la historia de los tratamientos dentales que se le haya realizado al paciente. En dicho gráfico se trabajará con un código de colores establecido para cada tratamiento, de tal forma que, en el momento en que el doctor ingrese el número de cédula de su paciente, se cargue el gráfico con los dientes y sus respectivos colores, según los tratamientos o intervenciones realizadas. Lo anterior facilita el trabajo al doctor puesto que no tiene que empezar a leer toda la historia para saber que se ha hecho en citas anteriores.

Otro punto en el que se observó posibilidad de mejora fue el del envío que realizan los doctores o las instituciones en los que los mismos trabajan, de las solicitudes de pago a las aseguradoras, el cual es un proceso tedioso para las dos partes puesto que implica el manejo de papelería y el transporte de la misma, y adicionalmente los analistas de las aseguradoras deben cargar manualmente toda esa información en su sistema para así poder empezar el proceso de validación de la data y proceder a autorizar o no los pagos.

Adicionalmente el transporte de dicha papelería implica un riesgo puesto que se puede perder y generar retrasos en los pagos y con ello la inconformidad de los doctores asociados con las aseguradoras.

Las aseguradoras se verán beneficiadas con la implementación de este sistema porque ayudará a evitar posibles fraudes por cobros indebidos en tratamientos o casos en los que se pretenda cobrar varias veces el mismo trabajo realizado.

El uso de cualquier estrategia a través de la cual los partícipes de una actividad pueden guiar el logro de sus objetivos con autonomía en el manejo de los recursos es conocido como Autogestión, este es un concepto al que las empresas vienen dando mayor importancia día a día.

Basados en estas necesidades de mejora de los procesos realizados por los profesionales de la salud en el área odontológica se concibió la realización de este sistema con el que se espera mejorar las condiciones de trabajo tanto de los doctores como de los analistas de las aseguradoras, del mismo modo lograr una mejora en la atención de los pacientes asegurados.

El sistema se compone de once módulos descritos a continuación:

**Módulo de pacientes:** permite llevar el control de los pacientes que podrán usar el servicio de salud odontológica de la institución, permitiendo agregar nuevos pacientes, modificarlos o eliminarlos.

**Módulo de doctores:** permite administrar los diferentes doctores de la institución, permitiendo agregar nuevos Doctores, modificarlos o eliminarlos.

**Módulo de tratamiento:** permite llevar el control de los tratamientos realizados al paciente, indicando descripción, el comienzo y fin de cada uno, controlando la cobertura máxima establecida que puede usar cada paciente por cada tratamiento durante la cobertura proporcionada por el contrato, esto en el caso de pacientes referidos por aseguradoras, permitiendo agregar nuevos tratamientos, modificarlos o eliminarlos. Podrá ver gráficamente todo lo que se le ha realizado al paciente a través de un gráfico manipulable, en el que podrá marcar en cada diente según un código de colores el tratamiento realizado.

**Módulo de enfermedades:** permite administrar las diferentes enfermedades y/o alergias que pueda tener el paciente. Cuenta con una sub-clasificación de enfermedades y permite agregar, modificar y eliminar según corresponda.

**Módulo de contratos:** permite llevar el control de los contratos de servicio firmados por la institución, indicando el comienzo y fin de cada contrato. Este módulo contiene contratos con las aseguradoras y contratos directos con los pacientes, permitiendo de manera eficiente y sencilla agregar nuevos contratos, modificarlos o eliminarlos.

**Módulo de pagos:** permite llevar el control de los pagos o abonos realizados por el paciente o la aseguradora correspondiente, pueden ser creados, modificados o anulados según corresponda. En cuanto a los abonos, se puede escoger a cual tratamiento se realizará dicho abono en caso de que tenga pendientes varios pagos.

**Módulo de aseguradoras:** permite llevar el control de las aseguradoras que tienen contrato vigente con la institución. El mismo controla que no se pueda exceder de los montos establecidos por tipo de tratamiento y ejerce la comunicación con las mismas, enviando toda la información necesaria para que las aseguradoras vean que tratamientos fueron realizados y ejecute los pagos. Este módulo ayuda a prevenir fraudes ya que con las formas manuales es más difícil controlar si ya se hizo un tratamiento o no a un determinado diente.

**Módulo de citas:** permite administrar las citas solicitadas por los pacientes y luego de ejecutada la misma genera la solicitud de pago para la aseguradora o el paciente según corresponda. Permite las operaciones básicas generar, modificar o eliminar.

**Módulo de reportes:** permite visualizar los diferentes reportes del sistema, con gráficos de barras o torta y mostrando los resultados de los mismos. Los reportes que posee el sistema son los siguientes: Citas por rango de fecha, tratamientos por paciente, pacientes por aseguradora, pagos realizados por tratamiento por cada paciente, contratos por paciente, contratos por aseguradora, pagos realizados por rango de fecha, enfermedades más frecuentes.

**Módulo de usuarios:** permite administrar los usuarios que podrán usar el sistema, permitiendo agregar, modificar y eliminar usuarios.

**Módulo de roles:** el módulo de seguridad, roles, permisos y gestión de usuarios de la aplicación web. El módulo de seguridad se encargará de asegurar la confidencialidad del intercambio de datos así como los niveles de visualización de los menús habilitados para los distintos tipos de usuarios. El acceso a la aplicación (autenticación), la creación de roles, asignación de permisos a los roles y la asignación de dichos roles a los usuarios, gestionar el registro, modificación, consulta y eliminación de los usuarios del sistema será regido por este módulo.

# CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

## Necesidades de la empresa

### I.1 Declaración de Oportunidad

Neo Solutions es una empresa joven e innovadora compuesta por profesionales altamente calificados en diversas áreas tecnológicas, cuya misión es "ser el socio tecnológico que permita el desarrollo e instalación de aplicaciones a sus clientes según sus necesidades, a través de las últimas tecnologías y sirviéndoles por encima de sus expectativas". Es por esto que se ha identificado la necesidad de desarrollar un sistema web para la administración del servicio de salud odontológica.

Actualmente algunos doctores o instituciones realizan la gestión y control de los pacientes, de forma manual, desde la verificación de la información, hasta las intervenciones y tratamientos que se realizan y los trámites administrativos para atender las solicitudes de los pacientes y los doctores. Existe la posibilidad como consecuencia de esto, que se generen retrasos e inconvenientes al momento de realizar todos los procesos y trámites administrativos, en especial con el pago que deben realizar las aseguradoras a la institución o al doctor por los servicios prestados a sus asegurados, ya que en la actualidad en los casos en que se realiza de forma manual, cada vez que se presenta un paciente a una cita, cada doctor o su asistente llena una forma en la que se especifica el tratamiento realizado ese día. Dichas formas se clasifican y guardan y al final de semana deben ser enviadas con un mensajero a la aseguradora correspondiente. Posteriormente, se debe esperar a que la aseguradora verifique cada una de las formas y las clasifique por paciente para evaluar y realizar la orden de pago.

Este manejo manual puede generar que los gastos del paciente se salgan de los límites establecidos por la cobertura ofrecida por el servicio de salud odontológica, además también deja la posibilidad de generación de fraudes y errores, ya que no se validan duplicados de facturas, ni si los médicos son familiares del paciente, entre otros.

## I.2 Solución Propuesta

En consecuencia al tiempo que consumen los procesos manuales que intervienen en el proceso actual de cobro y pago de servicios de los odontólogos, y la lentitud de dicho proceso, se propone desarrollar un sistema basado en una aplicación web para cumplir con todas las solicitudes y requerimientos especificados por Neo Solutions para la administración de los recursos del servicio de salud odontológica.

La intención de esta propuesta es garantizar una completa revisión de todos los procesos que intervienen en la forma de cobro y pago de los servicios prestados por los odontólogos a las diferentes aseguradoras, así como de los procesos administrativos de los consultorios y de este modo implementar el proceso de gestión de cobro que disminuya los tiempos de pagos y que cumpla y respete todas las condiciones del proceso que actualmente se ejecuta.

En Neo Solutions se ha decidido desarrollar una herramienta administrativa que le permita a las instituciones odontológicas y a las aseguradoras tener una completa y correcta gestión de sus recursos utilizados por los pacientes, y a su vez disminuya los tiempos de respuesta y garantizar que se cumplan todas las condiciones del servicio de todos sus pacientes y doctores asociados, la comunicación entre los doctores y las aseguradoras se realizará a través de un *Web Service*. Con este sistema el doctor podrá ver todos los tratamientos que se le han realizado al paciente a través de un gráfico manipulable, en el que podrá marcar cada uno de estos tratamientos, en cada pieza dental, según un código de colores preestablecido.

El sistema se dividirá en los módulos: “pacientes”, “doctores”, “tratamientos”, “enfermedades”, “contratos”, “pagos”, “aseguradoras”, “citas”, “reportes”, “usuarios” y “roles”; en varios de los cuales se realizará una integración con las

aseguradoras, con la finalidad de generar las solicitudes de pago por concepto de consultas, tratamientos, intervenciones quirúrgicas y demás gastos generados por la atención prestada al paciente.

El sistema permitirá llevar la administración total del consultorio o institución a la que pertenezca el doctor o doctores, podrán tener control de citas, llevar el historial de tratamientos realizados al paciente de manera gráfica y detallada, y permitirá llevar control de los pagos o abonos realizados por concepto de servicios y/o tratamientos prestados bien sea por parte del paciente o bien por parte de la aseguradora.

En cuanto al pago por parte de las aseguradoras, se evitará el envío de solicitudes físicas de pago por parte de los doctores o las instituciones, ya que dichas solicitudes se realizarán en línea y por esta misma vía las aseguradoras enviarán respuesta de las mismas. Con la esta interacción en línea podrán llevar control también del saldo restante de la cobertura correspondiente a cada asegurado evitando fraudes y posibles cobros injustificados.

Cada uno de los módulos que componen este aplicativo podrá realizar las operaciones básicas de ingresar, consultar, actualizar (en los casos que aplique) y eliminar (también en los casos que aplique), además de las funcionalidades propias de cada uno.

El sistema a desarrollar contempla el desarrollo de la interfaz del usuario, junto a las validaciones de datos correspondientes, reglas de negocio, colecciones de datos e integración del sistema de gestión de cobro y pago para garantizar un proceso sin papeles y controlado con el que se espera mejorar la eficiencia de los servicios prestados. El sistema aportará los siguientes cambios:

- Se espera generar mayor satisfacción en los asegurados ya que facilitará la atención por parte de los doctores o instituciones odontológicas, evitando al asegurado visitar las instalaciones de la empresa aseguradora en primera instancia para poder acudir a solicitar la atención.
- Con el sistema se busca mejorar el flujo de información de procesos referentes al cobro y pago de servicios prestados por parte de los doctores (odontólogos).
- Se busca generar satisfacción en las necesidades de los doctores (odontólogos) en el tiempo que lo requieran ya que ellos mismos podrán autogestionar sus peticiones de pagos.
- Se busca generar satisfacción al personal de las empresas aseguradoras, encargado de registrar las peticiones de pago de los doctores o las instituciones, ya que se automatizará el proceso de liquidación y se disminuirá la carga de trabajo mediante la integración de un *Web Service*, ya que no se tendrá que transcribir los datos de las peticiones, lo que se traduce en mejor operatividad para la empresa aseguradora.
- Se espera que el sistema contribuya a mejorar la atención a los doctores por parte de las aseguradoras, al poder ofrecer información en tiempo real de sus pagos mediante la consulta de pagos.
- Se espera que genere satisfacción en las aseguradoras porque ayudará a evitar fraudes, teniendo un control automatizado de los tratamientos realizados al asegurado se evitará posibles cobros dobles por el mismo tratamiento.

## I.3 Objetivo General

- Diseñar e implementar un sistema web para la administración del servicio de salud odontológica.

## I.4 Objetivos Específicos

- Diseñar e implementar la base de datos del sistema.
- Diseñar e implementar el módulo de pacientes.
- Diseñar e implementar el módulo de doctores.
- Diseñar e implementar el módulo de tratamientos.
- Diseñar e implementar el módulo de enfermedades.
- Diseñar e implementar el módulo de contratos.
- Diseñar e implementar el módulo de pagos.
- Diseñar e implementar el módulo de aseguradoras.
- Diseñar e implementar el módulo de citas.
- Diseñar e implementar el módulo de reportes.
- Diseñar e implementar el módulo de usuarios.
- Diseñar e implementar el módulo de roles.

## I.5 Aporte Funcional

- Rediseñar el proceso de control de los tratamientos e intervenciones realizadas al paciente, así como la validación y control de uso de los recursos médicos o quirúrgicos.
- Rediseñar el proceso de gestión de solicitudes en línea por parte de los doctores para la realización de solicitudes de pago a las aseguradoras.

## I.6 Aporte Tecnológico

- Evaluar la plataforma tecnológica (lenguaje de programación, manejador de base de datos, framework de desarrollo) y seleccionar la más adecuada para el desarrollo del aplicativo web.
- Diseñar e implementar una interfaz de integración y comunicación con las aseguradoras (al menos con una de ellas).

## I.7 Alcance

Se desarrollará un sistema informático conformado por once módulos que permitirán llevar el control administrativo del consultorio o de la institución y el historial médico de cada paciente de una manera eficiente y práctica. También incluirá un *Web Service* con el que se comunicarán los equipos informáticos de los doctores y los equipos de las aseguradoras para la transmisión de la información inherente a las solicitudes de pago por parte de los doctores y la respuesta que genera a dichas solitudes la(s) aseguradora(s), también se podrá saber en tiempo real el monto exacto que le que queda al paciente en su cobertura del seguro, evitando de este modo realizar tratamientos o intervenciones que estén fuera de cobertura.

El sistema contará con gestión de roles y usuarios, lo que dará seguridad al mismo, ya que se protege la información de los pacientes al no poder ser accedida por cualquier usuario. Debido al uso del sistema, se busca que las aseguradoras puedan llevar de manera más eficiente el control de los tratamientos por cada paciente y pueda evitar con este mayor control posibles cobros fraudulentos.

**Objetivos Específicos**

**Diseñar e implementar la base de datos del sistema.**

Diseñar e implementar la Base de Datos en donde se almacenarán los datos de la aplicación web. Se hará un diseño que se adapte a las necesidades que requiere el sistema para el manejo y persistencia de sus datos. Se creará, en primera instancia, un modelado de la data por medio de un modelo entidad-relación para representar las entidades relevantes del sistema y sus interrelaciones y propiedades. La misma deberá ser normalizada garantizando el uso óptimo de los recursos con los que cuente el doctor o la institución que hará uso del sistema.

**Diseñar e implementar el módulo de Pacientes.**

A partir de este módulo los doctores podrán llevar la información de los pacientes, datos de contacto, enfermedades, historial de tratamientos de forma gráfica y escrita, pagos y abonos realizados y cobertura disponible en el caso de que el paciente sea remitido por una aseguradora. Se permite agregar nuevos pacientes, modificarlos y eliminarlos. Para su diseño e implementación se deben cumplir las siguientes tareas:

1. Diseñar y validar la interfaz gráfica, buscando que sea amigable para el usuario final.
2. Diseñar e implementar un servicio web que consulte en la data de las aseguradoras la(s) póliza(s) de un titular y/o asegurado para saber la cobertura total y el saldo actual.
3. Diseñar e implementar un servicio web que actualice el saldo restante de la cobertura del titular y/o asegurado.

**Diseñar e implementar el módulo de Doctores.**

A partir de este módulo se podrá almacenar la información de los doctores, datos de contacto como son dirección, teléfono, email y su especialidad. Para su diseño e implementación se debe diseñar y validar una interfaz gráfica buscando que sea amigable para el usuario final.

**Diseñar e implementar el módulo de Tratamientos.**

Este módulo permite llevar el control de los tratamientos realizados al paciente, indicando descripción y el comienzo y fin de cada uno, controlando la cobertura máxima establecida que puede usar cada paciente por cada tratamiento durante la duración del contrato, esto en el caso de pacientes referidos por aseguradoras, permitiendo agregar nuevos tratamientos, modificarlos o eliminarlos. El doctor podrá ver gráficamente todo lo que se ha realizado al paciente a través de un gráfico manipulable, en el que podrá marcar en cada diente según un código de colores el tratamiento realizado. Para su diseño e implementación se deben cumplir las siguientes tareas:

1. Diseñar y validar la interfaz gráfica buscando que sea lo más amigable o agradable posible para el usuario final.
2. Diseñar e implementar el servicio que consulte la(s) póliza(s) de un titular y/o asegurado para saber la cobertura total y el saldo actual.
3. Diseñar e implementar el servicio que actualice el saldo restante de la cobertura del titular y/o asegurado.

**Diseñar e implementar el módulo de Enfermedades.**

Este módulo permite administrar las diferentes enfermedades y/o alergias que pueda tener el paciente. Cuenta con una sub-clasificación de enfermedades y permite agregar, modificar y eliminar según corresponda. Para su diseño e implementación se debe diseñar y validar una interfaz gráfica amigable para el usuario final.

**Diseñar e implementar el módulo de Contratos.**

Este módulo permite llevar el control de los contratos de servicio firmados por la institución, indicando el comienzo y fin de cada contrato. Este módulo contiene contratos con las aseguradoras y contratos directos con los pacientes, permitiendo agregar nuevos contratos, modificarlos o eliminarlos. Para su diseño e implementación se deben cumplir las siguientes tareas:

1. Diseñar y validar la interfaz gráfica buscando que sea amigable para el usuario final.
2. Diseñar e implementar un servicio que consulte la duración del contrato entre el doctor y la aseguradora, para de este modo no realizar atención a pacientes fuera del rango de fechas de duración del contrato. Este servicio permite validar que la fecha final que está almacenada en el sistema corresponde con la fecha final registrada en la compañía de seguros.
3. Las fechas de inicio y fin de los contratos directos con los pacientes serán almacenados directamente en BD.

**Diseñar e implementar el módulo de Pagos.**

Este módulo permite llevar el control de los pagos o abonos realizados por el paciente o la aseguradora correspondiente. Pueden ser creados, modificados o anulados según corresponda. En cuanto a los abonos, se puede escoger a cuál tratamiento se realizará dicho abono en caso de que tenga pendientes varios pagos. Para su diseño e implementación se deben cumplir las siguientes tareas:

1. Diseñar y validar la interfaz gráfica buscando que sea amigable para el usuario final.
2. Diseñar e implementar un servicio que realice la(s) petición(es) de pago a las aseguradoras.
3. Diseñar e implementar un servicio a través del cual la(s) aseguradora(s) enviarán su respuesta a la(s) petición(es) de pago a los doctores o instituciones.

**Diseñar e implementar el módulo de Aseguradoras.**

Este módulo permite llevar el control de las aseguradoras que tienen contrato vigente con el doctor o la institución odontológica. El mismo controla que no se pueda exceder de los montos establecidos por tipo de tratamiento y ejerce la comunicación con las empresas de seguros, enviando toda la información necesaria para que las aseguradoras vean que tratamientos fueron realizados y ejecute los pagos. Con el desarrollo de este módulo se busca ayudar a prevenir fraudes ya que en algunos casos es posible que con el uso de las formas manuales sea más difícil controlar si ya se hizo un tratamiento o no a un determinado diente.

**Diseñar e implementar el módulo de Citas.**

Este módulo permite administrar las citas solicitadas por los pacientes y luego de ejecutada la misma genera la solicitud de pago para la aseguradora o el paciente según corresponda. Permite generar, modificar o eliminar citas. El control de la fecha y hora se lleva a cabo con la API de Google Calendar.

**Diseñar e implementar el módulo de Reportes.**

Este módulo permite visualizar los diferentes reportes del sistema, mostrando los resultados de los mismos a través de tablas. Los reportes que poseerá el sistema son los siguientes:

- Citas por rango de fecha.
- Tratamientos por paciente.
- Pacientes por aseguradora.
- Pagos realizados por tratamiento por cada paciente.
- Contratos por Paciente.
- Contratos por Aseguradora.
- Pagos realizados por rango de fecha.
- Enfermedades más frecuentes.
- Algún otro que se crea conveniente.

**Diseñar e implementar el módulo de Usuarios.**

Este módulo permite administrar los usuarios que podrán usar el sistema, permitiendo agregar, modificar y eliminar usuarios por medio de una interfaz gráfica amigable.

**Diseñar e implementar el módulo de Roles.**

Desarrollar el módulo de roles, permisos y gestión de usuarios de la aplicación web. El módulo de seguridad se encargará de garantizar la confidencialidad del

intercambio de datos, así como los niveles de visualización de los menús habilitados para los distintos tipos de usuarios. El acceso a la aplicación (autenticación), la creación de roles, asignación de permisos a los roles y la asignación de dichos roles a los usuarios, gestionar el registro, modificación, consulta y eliminación de los usuarios del sistema será regido por este módulo.

**Alcance Aportes funcionales**

**Rediseñar el proceso de control de los tratamientos e intervenciones realizadas al paciente, así como la validación y control de uso de los recursos médicos o quirúrgicos.**

Actualmente, antes de ir a consulta, el paciente debe dirigirse a las oficinas de la aseguradora para solicitar una orden de servicio, que es la que solicitará el doctor al momento de la cita, y que es indispensable para poder atenderle. Para emitir dicha orden, en la aseguradora se encargan de verificar el estatus del plan de servicio del paciente y qué cobertura le queda para citas e intervenciones.

Con el rediseño del proceso se evitará que el paciente deba ir antes a la aseguradora, ya que con la solución propuesta el doctor podrá verificar en tiempo real el estatus del asegurado y generar en línea la orden de servicio. El sistema validará si es posible generar dicha orden o no, dependiendo de la cobertura y el tratamiento que se piense realizar, liberando a los analistas de la aseguradora de realizar dichas validaciones.

**Rediseñar el proceso de gestión de solicitudes en línea por parte de los doctores para la realización de solicitudes de pago a las aseguradoras.**

Actualmente, un mensajero debe llevar a cada una de las aseguradoras, todos los formatos que se llenaron de cada paciente atendido durante el período, para que en la aseguradora carguen dicha información, ocasionando esto el manejo de gran cantidad de papel y en muchos casos presentando inconsistencia entre la información que maneja el doctor y la que llega finalmente a la aseguradora ya que por ejemplo, si el motorizado pierde algún formato de algún paciente, el pago por parte de las aseguradora se retrasaría. A través del rediseño del proceso el doctor (o su asistente) será el responsable de cargar la información que se envía a las aseguradoras. Por otra parte, el doctor podrá consultar en línea sus pagos, transformándose así este proceso, en uno de autogestión de solicitudes en línea para el doctor.

**Alcance Aportes tecnológicos**

**Evaluar la plataforma tecnológica (lenguaje de programación, manejador de base de datos, framework de desarrollo) y seleccionar la más adecuada para el desarrollo del Portal web.**

Se tomará en cuenta hasta el más mínimo detalle del proyecto de tal forma de escoger la mejor opción y se trabaje con la tecnología adecuada para el desarrollo de la solución planteada.

**Diseñar e implementar una interfaz de integración y comunicación con las aseguradoras (al menos con una de ellas).**

Se diseñará una interfaz la cual ejercerá la comunicación con las aseguradoras a

través de un *web service.* Esta comunicación será fundamental para la realización de los pagos de forma oportuna.

Con esto, el proceso de verificación de la información recibida por la aseguradora se realizará de una forma más ágil, puesto que toda la información recibida será digital y estará ordenada y agrupada por paciente.

## I.8 Justificación

Esta solución ha sido desarrollada para cubrir las principales necesidades de las aseguradoras en cuanto a la interacción con sus asociados, así como para satisfacer y adaptarse a las necesidades de los doctores y/o instituciones odontológicas:

a. Un proceso de autogestión de solicitudes de pago para los doctores que permita también crear solicitudes de atención a los pacientes asegurados, cargar sus solicitudes y registrar y consultar sus pagos.
b. Una automatización en el proceso de evaluación de las solicitudes de pago, disminuirá la carga de trabajo en los analistas de las aseguradoras ya que estos no serán los encargados de cargar los datos de las solicitudes de pago de los doctores.
c. Disminución de tiempos en el proceso de pago, ya que las aseguradoras podrán dar respuesta de manera más ágil a las solicitudes de los doctores y llevarán mayor control de dichos pagos.
d. Disminución de tiempos en el proceso de análisis del historial de tratamientos del paciente, ya que en el sistema se verá reflejado en forma gráfica.

# CAPITULO II: MARCO TEÓRICO

En el presente capítulo, se detalla cada uno de los diferentes términos utilizados en la realización del Trabajo Instrumental de Grado.

## II.1. Tratamiento

En un contexto estrictamente médico, sin lugar a dudas el sentido del término que más se utiliza, el tratamiento será el conjunto de los medios de cualquier tipo, higiénicos, farmacológicos, quirúrgicos o bien físicos, los cuales tendrán como finalidad primaria la curación o el alivio de enfermedades o algunos síntomas de estas una vez que ya se ha llegado al diagnóstico de las mismas [24].

## II.2. Control de versiones

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de este modo se puedas recuperar versiones específicas durante un desarrollo [10].

## II.3. IDE (Entorno de Desarrollo Integrado)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes [13].

### II.3.1. Microsoft Visual Studio

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, servicios web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan

todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y hace más sencilla la creación de soluciones en varios lenguajes. Así mismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y servicios web XML [16].

### II.3.2. SharpDevelop

SharpDevelop es un entorno de desarrollo integrado libre para los lenguajes de programación C#, Visual Basic .NET, F#, Python, Ruby, Boo y C++. Es usado típicamente por aquellos programadores de los citados lenguajes, que no desean o no pueden usar el entorno de desarrollo de Microsoft, Microsoft Visual Studio [21].

## II.4. Base de Datos

Una base de datos es el conjunto de datos informativos organizados en un mismo contexto para su uso y vinculación. Se le llama base de datos a los bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre si algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto [5].

### II.4.1. Microsoft SQL Server

Es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos [15]. Algunas de sus características más importantes:

- Soporte de transacciones.
- Soporta procedimientos almacenados.

- Incluye también un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

### II.4.2. Navicat

Navicat es un administrador gráfico de base de datos y un software de desarrollo producido por PremiumSoft CyberTech Ltd. para MySQL, MariaDB, Oracle, SQLite, PostgreSQL y Microsoft SQL Server. Cuenta con un Explorador como interfaz gráfica de usuario soportando múltiples conexiones para bases de datos locales y remotos. Su diseño está pensado para satisfacer las diferentes necesidades de un amplio sector del público desde administradores y programadores de bases de datos a diferentes empresas que dan soporte y/o comparten información con clientes o socios [17].

## II.5. Framework

Un *framework* es un conjunto estandarizado de conceptos, prácticas y criterios usados para resolver problemas comunes de una manera rápida y eficiente [12].

### II.5.1. Microsoft .NET Framework

.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, independencia de plataforma y que permita un rápido desarrollo de aplicaciones. Basado en esta plataforma, Microsoft intenta desarrollar una estrategia horizontal que integre todos sus

productos, desde el Sistema Operativo hasta las herramientas de mercado [1].

### II.5.2. ASP .NET

ASP.NET es un modelo de desarrollo web unificado que incluye los servicios necesarios para crear aplicaciones web empresariales con el código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el *Common Language Runtime* (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J# [3].

### II.5.3. ASP .NET MVC Framework

Es un *framework* basado en ASP.NET, permite a los desarrolladores de software construir una aplicación web como una composición de tres funciones: modelo, vista y controlador [4].

### II.5.4. Entity Framework

*Entity* Framework (EF) es un marco de trabajo que implementa el patrón ORM (*object-relational* mapping) que permite a los desarrolladores de .NET trabajar con datos relacionales usando objetos específicos del dominio. Elimina la necesidad de la mayor parte del código de acceso a datos que los desarrolladores suelen tener que escribir [11].

### II.5.5. Twitter Bootstrap

Twitter Bootstrap, es un *framework* originalmente creado por Twitter Inc., que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de

adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como "responsive design" o diseño adaptativo [6].

## II.6. Lenguajes de Programación y de Consulta

### II.6.1. C#

Es un lenguaje de programación que se ha diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C [7].

### II.6.2. SQL

Las siglas SQL significan Structured Query Language, o su equivalente en español Lenguaje de Consulta Estructurado. Este es un lenguaje universal que está implementado en todos los motores de bases de datos razón por la cual es el lenguaje estándar de comunicación entre los diferentes motores existentes.

SQL es un lenguaje completamente normalizado que facilita el trabajo con cualquier tipo de lenguaje a la par con cualquier tipo de Base de Datos, sin embargo, esto no es equivalente a decir que es igual en todos los MBD, estos implementan diferentes funciones de acuerdo a la manera como más favorezca al MBD, estas funciones no siempre funcionan en otros [23].

## II.7. Servicio Web

Un servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. Es una máquina que atiende las peticiones de los clientes web y les envía los recursos solicitados. [20].

### II.7.1. SOAP

SOAP es un protocolo para el intercambio de mensajes sobre redes de computadoras, generalmente usando HTTP. Está basado en XML, a diferencia de DCOM y CORBA que son binarios; esto facilita la lectura por parte de los humanos, pero también los mensajes resultan más largos y, por lo tanto, considerablemente más lentos de transferir.

Existen múltiples tipos de modelos de mensajes en SOAP, pero, por lejos, el más común es el RPC, en donde un nodo de red (el cliente) envía un mensaje de solicitud a otro nodo (el servidor) y el servidor inmediatamente responde el mensaje al cliente.

Los mensajes SOAP, son independientes del sistema operativo, y pueden

transportarse en varios protocolos de internet como SMTP, MIME y HTTP. [22].

### II.7.2. REST

REST, REpresentational State Transfer, es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP.

REST permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que "entienda" HTTP, por lo que es increíblemente más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP y XML-RPC.

REST se definió en el 2000 por Roy Fielding, coautor principal también de la especificación HTTP. Podría considerarse REST como un *framework* para construir aplicaciones web respetando HTTP. Por lo tanto, REST es permite representar un tipo de arquitectura más natural y estándar para crear APIs para servicios orientados a Internet. [18].

# CAPÍTULO III: MARCO METODOLÓGICO

Este capítulo tiene como objetivo principal describir la metodología implementada en el desarrollo del proyecto. El uso de una metodología para el desarrollo de un sistema es de vital importancia ya que con ella se planifica y controla cada una de las fases del mismo; el uso de una metodología garantizará en la mayoría de los casos un sistema eficiente, eficaz y de buena calidad.

## III.1 Metodología RAD

El desarrollo rápido de aplicaciones o RAD (acrónimo en inglés de *rapid application development*) es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1980. El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE (*Computer Aided Software Engineering*). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución [14].

## III.2 Fases de la Metodología RAD

1. **Análisis y planificación de requerimientos**: se establecen los requerimientos y se definen las interacciones entre procesos, se realizan sesiones JAD (reuniones entre desarrolladores y usuarios finales).
2. **Diseño**: se lleva a cabo el diseño de un prototipo que será evaluado por los usuarios finales, se diseñan nuevos componentes y se definen los reusables. Dependiendo de los resultados de la evaluación, se realiza un refinamiento del prototipo y es evaluado nuevamente. Este proceso iterativo puede llevarse a cabo repetidas veces, pero se debe tener cuidado de no caer en un ciclo que retarde el desarrollo del proyecto.

3. **Construcción**: se realiza la construcción de prototipos. Desarrollo, pruebas y redefinición de requerimientos. Reuniones con el cliente para mostrar avances. El objetivo de esta fase es completar la funcionalidad del sistema.
4. **Implementación**: ésta es la fase de pruebas, entregas finales y elaboración de los manuales de usuario.

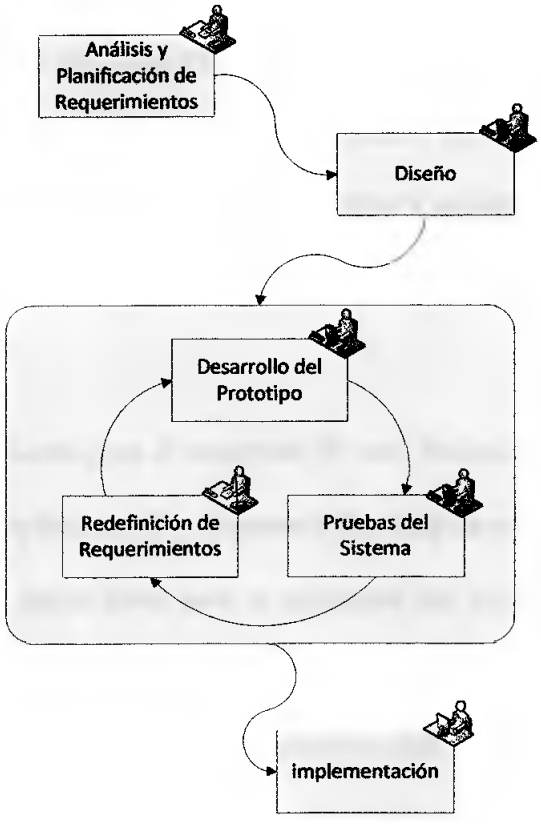



Figura 1: Fases de la metodología RAD
Fuente: Adaptación de https://curiosisimos.files.wordpress.com/2009/12/modelo-de-desarrollo-rapido-de-aplicaciones.pdf

## III.3 Características de la Metodología RAD

1. Aplicable a proyectos que duren entre 60 y 120 días de forma modular.
2. Las iteraciones no deben ser mayores a 3 semanas.
3. Involucra a los usuarios finales mediante sesiones JAD.
4. Recomienda un grupo pequeño de desarrolladores capaces de cumplir el rol de analistas, diseñadores y programadores.
5. Para el desarrollo del proyecto motiva el uso de herramientas CASE.
6. Obliga a la construcción de diversos prototipos (iterativos), en los cuales se puede apreciar el avance, ajustar requerimientos y aclarar dudas.
7. Propone dedicar un tiempo limitado a una tarea específica.

## III.4 Justificación

La metodología empleada para el desarrollo de este Trabajo Instrumental de Grado (TIG), RAD (*Rapid Application Development*), fue elegida porque las características así como sus fases la hacen ideal para la ejecución del presente TIG, permitiendo incluir a los usuarios finales (Doctor, Secretaria, Director) en el desarrollo del proyecto mediante sesiones JAD, ver el avance en cada iteración y establecer el uso de herramientas CASE para aumentar la productividad y reducir los tiempos en el desarrollo, también permite modularizar el proyecto que fue una de las características expresadas en la propuesta de este TIG. De igual modo, este proyecto fue establecido para una duración de desarrollo corta que se ajusta perfectamente a las características de la metodología RAD, con un alcance previamente definido, por lo que el uso de espacios limitados de tiempo (*time boxing*) es fundamental para evitar retrasos en el desarrollo del proyecto el cual será ejecutado por módulos.

Basado en las características de la metodología, el proyecto se dividió en doce (12) iteraciones (Tabla 1) detalladas en el capítulo IV Desarrollo.

Tabla 1: Iteraciones. Elaboración Propia

| | |
|---|---|
| 1 | Diseñar e implementar la base de datos. |
| 2 | Diseñar e implementar el módulo de pacientes. |
| 3 | Diseñar e implementar el módulo de doctores. |
| 4 | Diseñar e implementar el módulo de tratamientos. |
| 5 | Diseñar e implementar el módulo de enfermedades. |
| 6 | Diseñar e implementar módulo de contratos. |
| 7 | Diseñar e implementar la Interfaz de Integración y Comunicación con las Aseguradoras. |
| 8 | Diseñar e implementar el módulo de pagos. |
| 9 | Diseñar e implementar el módulo aseguradoras. |
| 10 | Diseñar e implementar el módulo de citas. |
| 11 | Diseñar e implementar el módulo de reportes. |
| 12 | Diseñar e implementar el módulo de usuarios y roles. |

# CAPÍTULO IV: DESARROLLO

El presente capítulo describe paso a paso la ejecución del proyecto, a partir de la aplicación de los instrumentos teóricos y metodológicos elegidos. Antes de empezar las iteraciones de desarrollo se realizó una investigación preliminar, así como un análisis y diseño de los requerimientos y procesos que conforman los objetivos de este proyecto:

## IV.1 Investigación Preliminar

Se investigaron las herramientas tecnológicas, los distintos métodos para el desarrollo del sistema y la preparación del ambiente de desarrollo. Se realizó una evaluación y selección entre las herramientas que facilitarán el desarrollo del proyecto.

### IV.1.1 Estudio de herramientas tecnológicas

Las herramientas tecnológicas a evaluar eran las siguientes:

- Dos IDE de desarrollo para el diseño y programación de aplicaciones, se seleccionará uno entre Microsoft Visual Studio y SharpDevelop.
- Herramienta de desarrollo para base de datos a escoger una entre Navicat, Toad, My SQL Worbench.
- Manejador de base de datos entre Microsoft SQL Server y My SQL.
- Plantilla web desarrollada con el framework ASP.Net MVC.
- Entity Framework.

### IV.1.2 Evaluación y selección de la herramienta de desarrollo

Para este proyecto uno de los objetivos planteados es evaluar la plataforma de desarrollo y seleccionar la más idónea para el desarrollo del aplicativo web.

Las herramientas de entorno de desarrollo integrado (IDE) poseen características propias que se deben evaluar para el desarrollo de un proyecto, para determinar el entorno que mejor se adapta a las necesidades del desarrollo han sido seleccionados distintos criterios técnicos necesarios para el desarrollo de la aplicación.

Tabla 2: Evaluación de Herramienta de Desarrollo.
Fuente: *SharpDevelop Community*

| ***Feature*** | ***SharpDevelop*** | ***Visual Studio Express Edition*** |
|---|---|---|
| *Code Auto-Completion* | *Yes* | *Yes* |
| *Code Syntax Highlighting* | *Yes* | *Yes* |
| *Windows Forms Designer* | *Yes* | *Yes* |
| *Web Forms Designer* | *No* | *Provided with Visual Web Developer* |
| *WPF Designer* | *Yes* | *Yes* |
| *Code Coverage* | *Yes* | *No* |
| *Unit Testing* | *Yes* | *No* |
| *Languages Supported* | *C#, C++, VB.NET, Boo, F#, IronPython, IronRuby* | *C#, C++, VB.NET, JavaScript (IronPython and IronRuby support is available with Visual Studio 2010 Shell)* |
| *Window Phone Development Support* | *No* | *Yes* |
| *Cloud Application Development (Windows Azure)* | *No* | *Yes* |
| *Profiling* | *Yes* | *No* |
| *Help Documentation* | *No* | *Yes* |
| *Plug-in Support* | *Yes* | *No* |
| *Insert PInvoke Signatures* | *Yes* | *No* |
| *Testing Regular Expressions* | *Yes* | *No* |
| *Class View* | *Yes* | *Yes* |
| *Solution Explorer* | *Yes* | *Yes* |
| *Project and Solution File Format* | *MSBuild* | *MSBuild* |
| *Web References* | *Yes* | *Yes* |
| *Refactorings* | *Rename, Extract Method, Remove unused imports* | *Rename, Extract Method* |

| | | |
|---|---|---|
| *Go To Definition* | *Yes* | *Yes* |
| *Find References* | *Yes* | *Yes* |
| *Code Generation* | *Yes. Not as powerful as Visual Studio's Code Snippet Manager.* | *Yes* |
| *Object Browser* | *No* | *Yes* |
| *Database Explorer* | *Yes* | *Yes* |
| *Publishing* | *No* | *Yes* |
| *Data Sources View* | *No* | *Yes* |
| *Add Data Source Wizard* | *No* | *Yes* |
| *Document Outline View* | *No* | *Yes* |

En relación al cuadro comparativo (Tabla 2) de las herramientas de desarrollo Microsoft Visual Studio y SharpDevelop se aprecia que los dos cumplen con las características mínimas necesarias para el desarrollo del sistema pero es Visual Studio el elegido por ofrecer más características como la documentación de ayuda *online* y *outline*, el diseñador de formas web y un generador de código más potente.

### IV.1.3 Evaluación y selección de la herramienta de desarrollo de la base de datos

Se busca que la herramienta de desarrollo de la base de datos sea un administrador gráfico multiplataforma y cuente con una interfaz gráfica de usuario soportando múltiples conexiones para bases de datos locales y remotas.

Tabla 3: Evaluación de Herramienta de Desarrollo de Base de Datos.
Fuente: http://desarrolloweb.blogspot.com

| Herramienta | Win | Linux | Oracle | My SQL | Postgre SQL | SQL Server | Lenguaje de Prog. |
|---|---|---|---|---|---|---|---|
| ABIS-Query | ✓ Sí | ✗ No | ✓ Sí | ✓ Sí | ✓ Sí | ✓ Sí | C# |
| Data Analyst Toolkit | ✓ Sí | ✗ No | ✓ Sí | ✓ Sí | ✓ Sí | ✓ Sí | C# |
| Database Master with LINQ | ✓ Sí | ✗ No | ✓ Sí | ✓ Sí | ✓ Sí | ✓ Sí | .Net / C# |
| dbForge SQL Complete | ✓ Sí | ✗ No | ✗ No | ✗ No | ✗ No | ✓ Sí | C# |
| Microsoft SQL Server Management Studio | ✓ Sí | ✗ No | | | | ✓ Sí | .Net |
| MySQL Workbench | ✓ Sí | ✓ Sí | | ✓ Sí | | | C++/C# Obj ective-C |
| Navicat | ✓ Sí | ✓ Sí | ✓ Sí | ✓ Sí | ✓ Sí | ✓ Sí | C++/C# Obj ective-C |
| Toad | ✓ Sí | ✗ No | ✓ Sí | ✓ Sí | | ✓ Sí | Embarcade ro Delphi, C#. NET |

En relación al cuadro comparativo (Tabla 3) de las herramientas de desarrollo de base de datos se aprecia que todas cumplen con las características mínimas necesarias para el desarrollo del sistema pero es Navicat la elegida por ofrecer más características:

- ✓ Analizar, repasar y reparar tablas
- ✓ Administrador Visual para la gestión de usuarios y sus privilegios
- ✓ Programar copias de seguridad
- ✓ Importar y Exportar datos desde MS Excel y MS Access
- ✓ Copia de seguridad de bases de datos y scripts de SQL
- ✓ Ejecutar scripts SQL
- ✓ Múltiples editores de SQL con sintaxy resaltada
- ✓ Sincronización de Datos y Estructuras
- ✓ con capacidad para arrastrar y colocar tablas
- ✓ Crear consultas de parámetros
- ✓ Modelado de datos
- ✓ Vista de Diagrama ER
- ✓ Amplia búsqueda de base de datos
- ✓ Minify SQL
- ✓ Navicat Cloud
- ✓ Sincronizar conexiones
- ✓ Sincronizar consultas
- ✓ Sincronizar modelos
- ✓ Sincronizar grupos virtuales

### IV.1.4 Evaluación y selección del Framework para el Desarrollo Web

Un framework para aplicaciones web es un framework diseñado para apoyar el desarrollo de sitios web dinámicos, aplicaciones web y servicios web. Este tipo de frameworks intenta aliviar el exceso de carga asociado con actividades comunes usadas en desarrollos web.

Tabla 4: Evaluación de Framework para Desarrollo Web.
Fuente: http://www.w3ii.com/es/mvc_framework/mvc_framework_asp.net_forms.html

| Factores de Comparación | Web Forms | ASP.NET MVC |
|---|---|---|
| Enfoque de representación | Sigue el enfoque del patrón de control de página para renderizar el diseño | Se utiliza el método de control frontal |
| Separación de intereses | No hay separación de intereses y todas las formas web están estrechamente acopladas | Separación muy clara de intereses |
| Pruebas automatizadas | Las pruebas automatizadas son realmente difíciles | TDD es bastante simple en MVC |
| Rendimiento | Lento debido a gran vista de estado | Rápido debido al enfoque limpio y no hay vista de estado |
| Ciclo de vida | El modelo de formularios web sigue el ciclo de vida de página | No hay ciclo de vida de página como con los formularios web |
| Controles | Muchos controles secundarios del servidor | Sin salir del cuadro de control. Se pueden usar controles de terceros |
| Control sobre el diseño | La abstracción anterior era buena pero proporciona un control limitado sobre HTML, JavaScript y CSS que es necesario en muchos casos | Control total sobre HTML, JavaScript y CSS |
| Escalabilidad | Es buena para aplicaciones a pequeña escala con un equipo limitado | Es mejor como enfoque recomendado para aplicaciones a gran escala |

En relación al cuadro comparativo (Tabla 4) de los Framework se aprecia que tanto Web Forms como ASP .NET MVC cumplen con las características necesarias para el desarrollo del sistema pero es ASP el elegido por ofrecer mejores características como la separación clara de intereses y el rendimiento más rápido.

La plataforma de desarrollo quedó conformada por las siguientes herramientas y lenguajes de programación mostrados a continuación:

**Herramientas**

- IDE Microsoft Visual Studio: herramienta de desarrollo seleccionada.
- Base de datos Microsoft SQL Server: Ofrece una potente forma de unir SQL e Internet.
- Navicat: es un programa de mantenimiento y consulta de bases de datos, permite de manera cómoda la gestión de las mismas.

**Lenguajes de programación (*Front-End*)**

- HTML5: por ser la última versión del lenguaje universal para diseño de aplicaciones web (HTML).
- CSS3: por ser la última versión e incluir nuevas funcionalidades del lenguaje de definición de estilos de un documento (CSS).
- Razor HTML: es un motor de vistas que busca reducir la cantidad de código necesario para crear las mismas, fácil de aprender y que funciona con cualquier editor de texto.
- jQuery: se pueden agregar *plugins* fácilmente, traduciéndose esto en un ahorro substancial de tiempo y esfuerzo.

**Lenguajes de programación (Back-End)**

- SQL: por ser el lenguaje universal para la creación y manipulación de las tablas (estructura) de bases de datos relacionales.
- C#: es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

**Frameworks**

- Twitter Bootstrap: es un framework que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como "responsive design" o diseño adaptativo.
- Microsoft .Net: es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.
- ASP.Net MVC: El marco de ASP.NET MVC proporciona una alternativa al modelo de formularios Web Forms de ASP.NET para crear aplicaciones web. El marco de ASP.NET MVC es un marco de presentación de poca complejidad y fácil de comprobar que (como las aplicaciones basadas en formularios Web Forms) se integra con las características de ASP.NET existentes, tales como páginas maestras y la autenticación basada en pertenencia.
- Entity Framework: es una tecnología desarrollada por Microsoft, que a través de ADO.NET genera un conjunto de objetos que están directamente ligados a una Base de Datos, permitiendo a los desarrolladores manejar dichos objetos en lugar de utilizar lenguaje SQL contra la Base de Datos.

## IV.2 Análisis, Planificación de Requerimientos y Diseño

### IV2.1. Análisis y Planificación de requerimientos

En esta parte el objetivo es definir las necesidades o requerimientos del cliente con base en la propuesta entregada y al levantamiento de información. En esta sección se deben obtener las características de la solución y el alcance del proyecto para producir los módulos que permitan satisfacer los requerimientos y los objetivos del proyecto, para esto se cuenta con el apoyo de los Casos de Uso (Apéndice A).

| Requerimiento | Descripción del requerimiento |
|---|---|
| Funcional | Consultar, crear, modificar y eliminar Pacientes. |
| | Consultar, crear, modificar y eliminar Tratamientos. |
| | Consultar, crear, modificar y eliminar Enfermedades. |
| | Consultar, crear, modificar y eliminar Doctores. |
| | Consultar, crear, modificar y eliminar Contratos. |
| | Consultar, crear, modificar y eliminar Aseguradoras. |
| | Consultar, crear, modificar y eliminar Citas. |
| | Consultar, crear, modificar y eliminar Pagos. |
| | Consultar, crear, modificar y eliminar Reportes. |
| | Consultar, crear, modificar y eliminar Usuarios. |
| | Consultar, crear, modificar y eliminar Roles. |
| | Enviar solicitudes de pago a las Aseguradoras. |
| | Recibir respuesta de las Aseguradoras por las Solicitudes enviadas. |
| | Ofrecer una interfaz agradable al usuario. |
| | El sistema debe ser escalable y permitir futuros desarrollos de nuevas funcionalidades y un fácil mantenimiento. |
| | Manuales de usuario. |
| | El sistema debe ser desarrollado y entregado por módulos. |

Tabla 5: Lista de requerimientos del proyecto.
Fuente: elaboración propia

Basado en los requerimientos funcionales se determinaron los actores que interactuarían con el sistema (Ver Tabla 6).

| Tipo de actor | Descripción |
|---|---|
| Doctor | Usuario creado a los Doctores para que accedan al Sistema. |
| Secretaria | Usuario creado a la Secretaria para que acceda al Sistema. |
| Director | Usuario creado al Director para que acceda al Sistema. |

Tabla 6: Listado de actores planteados en el sistema.
Fuente: Elaboración Propia

### IV.2.1 Diseño

La arquitectura del sistema desarrollado es orientada a servicios. Permite la creación de sistemas de información altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

### IV 2.2.1 Rediseñar Proceso de Gestión de Pagos

Se rediseñó el proceso de gestión de pagos, ahora permite a los doctores crear solicitudes de pago a las aseguradoras a través del sistema, con el proceso existente las solicitudes se realizan de forma manual, el envío que realizan las instituciones o doctores de dichas solicitudes es tedioso para las dos partes (doctores y aseguradoras) puesto que implica el manejo de papelería y el transporte de la misma, adicionalmente los analistas de las aseguradoras deben cargar manualmente toda esa información en su sistema para así poder empezar el proceso de validación de la data y proceder a autorizar o no los pagos.

Adicionalmente el transporte de dicha papelería implica un riesgo puesto que puede perderse y generar retrasos en los pagos y con ello la inconformidad de los doctores asociados con las aseguradoras.

Se busca que las aseguradoras se vean beneficiadas con la implementación de este nuevo proceso porque ayudará a evitar posibles fraudes por cobros indebidos en tratamientos o casos en los que se pretenda cobrar varias veces el mismo trabajo realizado, de igual modo se evitará la transcripción manual por parte de los analistas.

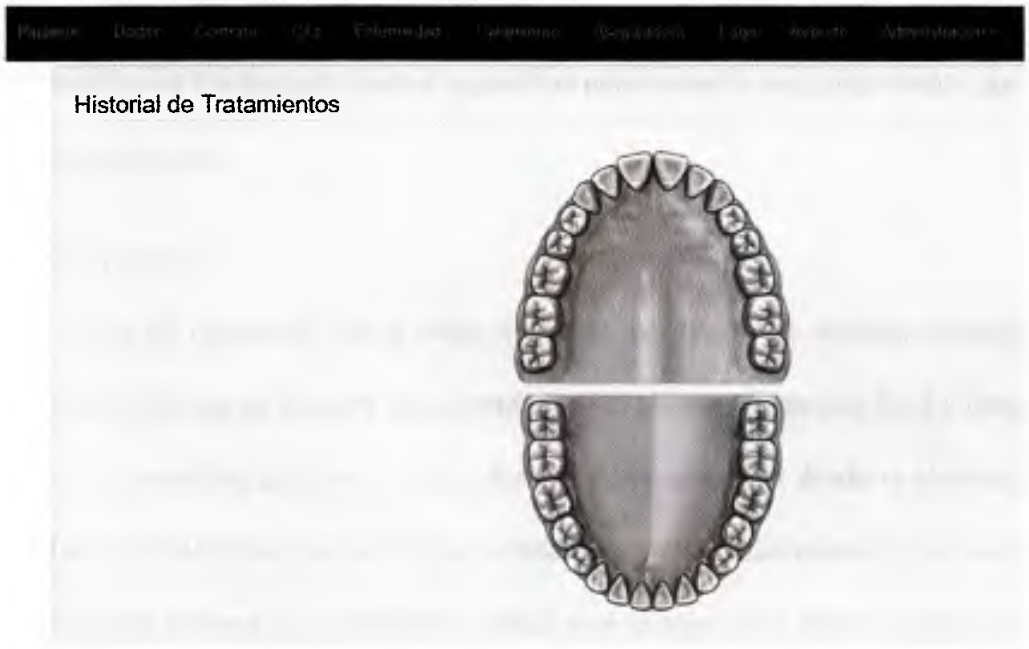
De esta manera los doctores son los responsables de autogestionar sus cobros a las aseguradoras y cargarlos al sistema, y los analistas de las aseguradoras sólo se dedicarán a analizar cada solicitud de pago perteneciente a cada doctor y determinar si se aprueba o no su pago.

En el diseño del proceso de autogestión se puede observar un rediseño en el proceso de creación de solicitudes de pago, donde ya el doctor asociado o el encargado del proceso en la institución no tienen la necesidad de asistir a las instalaciones de la aseguradora para que llevar las solicitudes de pago.

### IV 2.2.2 Rediseñar Proceso de Control de Tratamientos

Se rediseñó el proceso de control de tratamientos que permite a los doctores o sus asistentes cargar en el sistema en tiempo real el tratamiento que se está realizando al paciente, del mismo modo permite a las aseguradoras llevar un mejor control de dichos procedimientos ya que se llevaba de forma manual, a la aseguradoras llegaban los formatos de los tratamientos por cada paciente de forma física y los analistas debían registrar en su sistema estos tratamientos, sin embargo en conversaciones con los usuarios se evidenció que en muchos casos se podía repetir la carga de un mismo tratamiento en los sistemas de las

aseguradoras y generar cobros múltiples para un mismo procedimiento en un mismo paciente, por ejemplo en un caso en particular se llegó a realizar el pago de la extracción de un mismo diente en tres ocasiones, con el rediseño las aseguradoras tendrán digitalizados todos los tratamientos y contarán con la nueva funcionalidad del historial del paciente de forma gráfica gracias al uso del gráfico dental.




## IV.3 Desarrollo

El desarrollo del sistema se realizó bajo las herramientas descritas en la sección anterior. Para el desarrollo de la interfaz gráfica de los módulos del sistema se utilizaron componentes tipo plantilla de Twitter Bootstrap y se incorporó el uso de la librería jQuery que permite crear tablas dinámicas, ya sea con contenido embebido en la misma página HTML o con contenido dinámico

extraído de un servidor para mostrar las consultas.

El sistema se desarrolló mediante las siguientes 12 iteraciones.

### IV.3.1 Iteración 1: Diseñar e implementar la base de datos en donde se almacenarán los datos del sistema.

#### IV.3.1.1 Análisis y planificación de requerimientos

En reunión con el tutor empresarial y usuarios del sistema se definieron las especificaciones requeridas para el diseño y desarrollo de la base de datos, se identificaron los requerimientos necesarios para cumplir con cada detalle que necesite el sistema.

#### IV.3.1.2 Diseño

Para el desarrollo de la base de datos se diseñó el modelo entidad relación (ER) que se basa en las necesidades del cliente (Apéndice B). La base de datos permitirá guardar y manipular los datos enviados desde el sistema. La base de datos contiene las tablas necesarias para el funcionamiento de cada módulo del sistema. Las principales tablas que componen la base de datos del sistema son las siguientes:

- Pacientes: almacena los datos básicos del paciente como son nombre, apellido, cédula, teléfono, dirección.
- Tratamientos: almacena el detalle de cada uno de los tratamientos realizados por el doctor o la institución.
- Doctores: almacena los datos básicos del doctor, como son nombre, apellido, cédula, teléfono.
- Dientes: almacena la información básica de las piezas dentales.

- Trata_diente: guarda la relación entre las tablas Tratamiento y Diente.
- Historial: Lleva la relación entre las tablas Paciente, Diente y Doctor.
- Detalle_historial: guarda la relación entre las tablas Historial y Trata_diente, así como una descripción.
- Aseguradora: almacena los datos básicos de cada aseguradora como son nombre, apellido, cédula, teléfono, dirección.
- Paciente_aseguradora: guarda la relación entre las tablas Paciente y Aseguradora, de esta manera se puede determinar cuál es la empresa aseguradora del paciente.
- Contrato: almacena los contratos existentes entre doctor y paciente o doctor y aseguradora.
- Cita: almacena la fecha de la cita, el paciente y el doctor.
- Cita_tratamiento: guarda la relación entre la cita y el tratamiento que se realizó.

#### IV.3.1.3 Construcción

Una vez diseñado el modelo entidad relación (ER) se procedió a realizar los scripts para crear las tablas definidas en la base de datos SQL Server. Para el sistema fue de gran importancia el diseño y construcción de la base de datos ya que en ella se almacenan los datos y la lógica de negocios utilizada en el sistema.

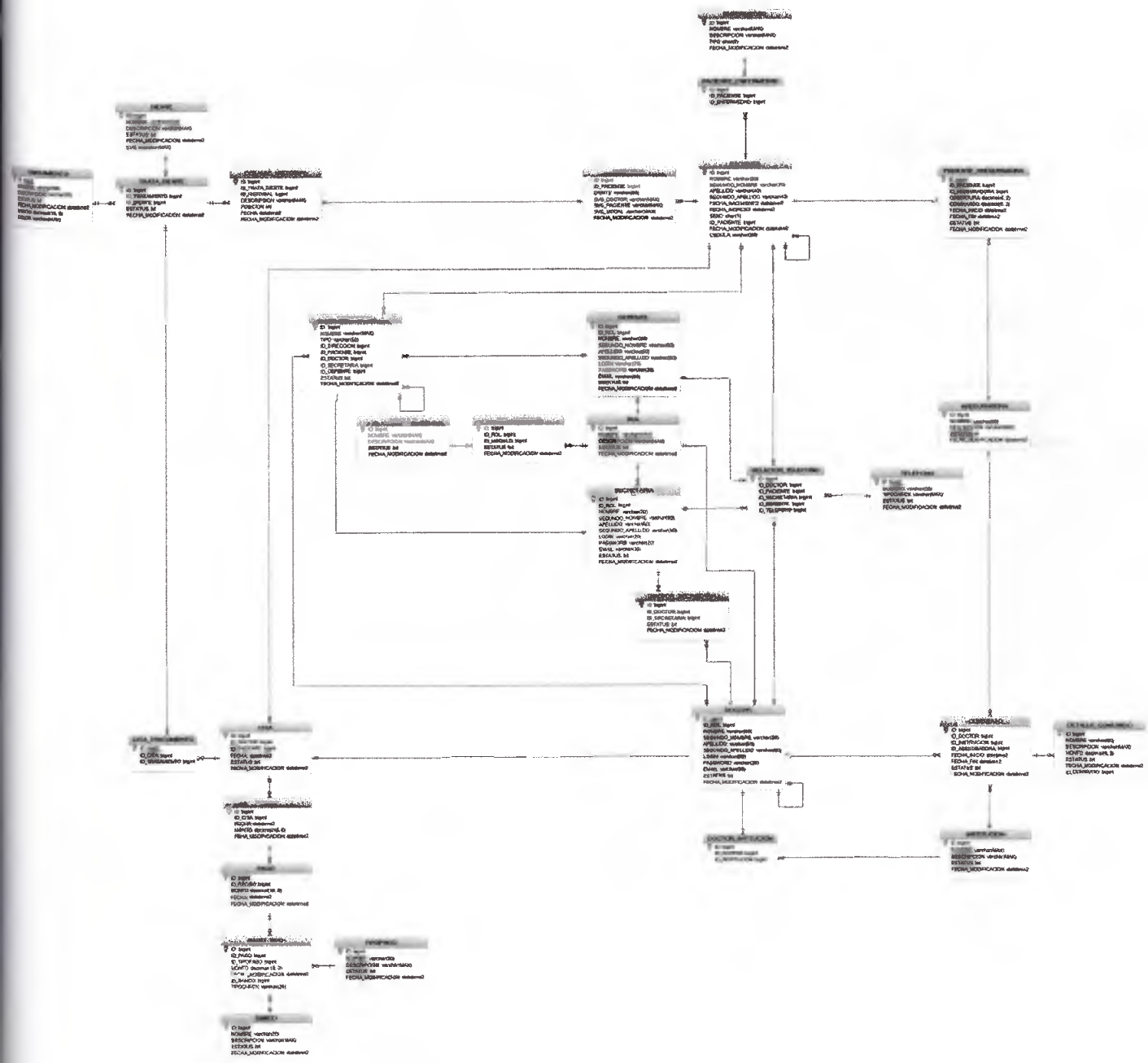


Figura 10: Modelo ER
Fuente: Elaboración Propia

#### IV.3.1.4 Implementación

En esta etapa del diseño e implementación de la base de datos, se realizó una reunión con el tutor empresarial y se realizaron las siguientes actividades:

- Revisión del modelo entidad relación.
- Revisión de normalización del modelo entidad relación.
- Revisión de estándares de diseño (nombres de tablas, columnas, *constraints*), adaptándolo al estándar de la empresa (Neo Solutions).
- Ejecución de operaciones de prueba de manipulación de datos en cada una de las tablas.

### IV.3.2 Iteración 2: Diseñar e implementar módulo de pacientes.

#### IV.3.2.1 Análisis y planificación de requerimientos

El análisis y planificación de requerimientos de este módulo se basó en el diagrama de caso de uso (Apéndice A.1) y las siguientes preguntas:

1. ¿Qué información se necesita del paciente?

   La respuesta fue la cédula de identidad, nombre, apellido, sexo, dirección, teléfono de contacto.

2. ¿Qué se debe validar al momento de crear un paciente?

   La respuesta fue que el paciente no estuviese registrado con anterioridad, que no se registren tratamientos repetidos para la misma causa y tomando en cuenta la estructura de los datos a registrar.

Se debe tener en cuenta que en la ficha del paciente debe poderse registrar las alergias y enfermedades que el mismo padece o padeció, así como el estado de los dientes al momento de hacer la primera evaluación del paciente en esta institución o consultorio.

Por otra parte, se estableció la necesidad de un cuadro de búsqueda inteligente que vaya descartando pacientes a medida que se escribe en él hasta quedar con el paciente o los pacientes que tengan contenida la palabra escrita en el cuadro de texto.

#### IV.3.2.2 Diseño

Para el desarrollo de este módulo se diseñó un prototipo de la interfaz gráfica (Apéndice E) que permitiera crear paciente, consultar, modificar toda

su información y eliminar si se requiere, se diseñaron cuatro pantallas para realizar estos requerimientos, la pantalla de nuevo paciente, en la que se asigna nombres, apellidos, información de contacto, también se puede asociar diferentes enfermedades y/o alergias así como los tratamientos que tiene con anterioridad; una en la que se listan todas los pacientes que están registrados, listados en páginas de 10 registros para facilitar al usuario la navegación por el sistema; una pantalla de consulta (Apéndice E.1) y modificación en la que se puede ver en detalle uno a uno los pacientes que se quiera y hacer las respectivas modificaciones y registro de nuevos tratamientos realizados; y una para hacer eliminación (Apéndice E.2) si así se requiere.

Se realizó este prototipo adecuándolo en cada iteración hasta que fue aprobado por el tutor empresarial y los usuarios que lo utilizarían.

Para el histórico de los tratamientos de cada paciente se realizó un gráfico dental (Apéndice H.4) en el que se puede seleccionar diente a diente y marcar el tratamiento realizado, para esto cada diente tiene una división de tres por tres (3x3), en cada una de estas nueve (9) divisiones se puede elegir un tratamiento.

El gráfico dental se desarrolló a partir de gráficos vectoriales redimensionables (SVG), el manejo de vectores permite crear imágenes complejas. Dicho prototipo también fue aprobado por el tutor empresarial.

#### IV.3.2.3 Construcción

Una vez aprobado el prototipo de la interfaz gráfica, el prototipo del gráfico dental y definida la funcionalidad del módulo y los procesos que

intervienen en la creación y consulta de pacientes se procede a la construcción del módulo definido a continuación por el desarrollo de los siguientes procesos:

1. Crear, consultar, modificar y eliminar paciente.
2. Consulta de enfermedades del paciente.
3. Seleccionar enfermedades nuevas al paciente.
4. Consulta de alergias del paciente.
5. Seleccionar alergias nuevas al paciente.
6. Seleccionar Tratamientos para el paciente.

Adicionalmente para el gráfico dental se utilizaron gráficos vectoriales redimensionables (del inglés *Scalable Vector Graphics*) o SVG, éstos son una especificación para describir gráficos vectoriales bidimensionales, tanto estáticos como animados en formato XML.

#### IV.3.2.4 Implementación

En esta etapa se crearon diferentes pacientes con diferentes enfermedades y alergias, así como diferentes tratamientos asignados. Se comprobó que las validaciones funcionan correctamente, se validaron las reglas de negocio incluidas en el módulo, entre las que destacan no poder repetir tratamientos en el mismo diente, es decir, que no se pudiera solicitar un mismo tipo de tratamiento para la misma zona afectada (Por ejemplo, no se puede extraer dos veces la misma pieza dental).

### IV.3.3 Iteración 3: Diseñar e implementar módulo de doctores.

#### IV.3.3.1 Análisis y planificación de requerimientos

El análisis y planificación de requerimientos de este módulo se basó en el diagrama de caso de uso (Apéndice A.2) y las siguientes preguntas:

1. ¿Qué información se necesita del doctor?

   La respuesta fue la cédula de identidad, nombre, apellido, sexo, el rol que desempeña, datos de contacto.

2. ¿Qué se debe validar al momento de crear un doctor?

   La respuesta fue que el doctor no estuviese registrado con anterioridad.

Se debe tener en cuenta que en la ficha del doctor debe poder registrarse los contratos en caso de que existan con las empresas aseguradoras, teniendo en cuenta la fecha de inicio y fecha de fin de dicho contrato. Se debe validar que no se registren nuevos contratos con aseguradoras que aún tenga vigente alguno, en caso tal se puede modificar la fecha de fin del contrato vigente.

También se estableció la necesidad de un cuadro de búsqueda inteligente que vaya descartando de la lista a los doctores a medida que se escribe en él hasta quedar con el doctor o los doctores que tengan contenida la palabra escrita en el cuadro de texto.

#### IV.3.3.2 Diseño

Para el desarrollo de este módulo se diseñó un prototipo de la interfaz gráfica (Apéndice F) que permitiera crear doctor, consultar, modificar toda su información y eliminar si se requiere, se diseñaron cuatro pantallas para

realizar estos requerimientos, la pantalla de crear doctor (Apéndice F.1), en la que se asigna nombres, apellidos, información de contacto; una en la que se listan todas los doctores que están registrados, listados en páginas de 10 registros para hacer amigable al usuario la navegación por el sistema; una pantalla de consulta (Apéndice F.2), modificación (Apéndice F.3) en la que se puede ver en detalle uno a uno los doctores que se quiera y modificar lo que sea necesario; y una para hacer eliminación (Apéndice F.4) si así se requiere.

En este módulo también se puede asociar las aseguradoras con las que tiene contrato. Se realizó este prototipo adecuándolo en cada iteración hasta que fue aprobado por el tutor empresarial y los usuarios.

#### IV.3.3.3 Construcción

Una vez aprobado el prototipo de interfaz gráfica (Apéndice F) y definida la funcionalidad del módulo y los procesos que intervienen en la creación y consulta de doctores se procede a la construcción del módulo definido a continuación por el desarrollo de los siguientes procesos:

1. Crear, consultar, modificar y eliminar doctor.
2. Consulta de rol del doctor.
3. Seleccionar nuevo rol del doctor.
4. Consultar las asociaciones con las aseguradoras.
5. Seleccionar Aseguradoras nuevas al doctor.

#### IV.3.3.4 Implementación

En esta etapa se crearon diferentes doctores con diferentes convenios con aseguradoras. Se comprobó que las validaciones funcionan correctamente, se validaron las reglas de negocio incluidas en el módulo, entre las que destacan

no poder repetir la creación de un doctor ni repetir los convenios con las aseguradoras.

### IV.3.4 Iteración 4: Diseñar e implementar módulo de tratamientos.

#### IV.3.4.1 Análisis y planificación de requerimientos

El análisis y planificación de requerimientos de este módulo se basó en el diagrama de caso de uso (Apéndice A.6) y las siguientes preguntas:

1. ¿Qué información se necesita del tratamiento?

   La respuesta fue nombre, descripción, color (este color será el que identifique al tratamiento en el gráfico dental), monto (costo) y la lista de los dientes que pueden ser intervenidos con el mismo.

2. ¿Qué se debe validar al momento de crear un tratamiento?

   La respuesta fue que el tratamiento no estuviese registrado con anterioridad.

#### IV.3.4.2 Diseño

Para el desarrollo de este módulo se diseñó un prototipo de la interfaz gráfica (Apéndice J) que permitiera crear tratamiento, consultar, modificar toda su información y eliminar si se requiere, se diseñaron cuatro pantallas para realizar estos requerimientos, la pantalla de nuevo tratamientos (Apéndice J.1), en la que se asigna nombre, descripción, color (este color será el que identifique al tratamiento en el gráfico dental), monto (costo) y la lista de los dientes que pueden ser intervenidos con el mismo; una en la que se listan todas los tratamientos que están registrados, listados en páginas de 10 registros para hacer amigable al usuario la navegación por el sistema; una pantalla de modificación (Apéndice J.2) en la que se puede ver en detalle uno a uno los tratamientos que se quiera y modificar si es necesario y una para

hacer eliminación (Apéndice J.3) si así se requiere.

En este módulo también se puede asociar los tratamientos para cada diente o para todos si así se requiere, se asigna un color con el que se podrá identificar fácilmente en el gráfico dental que tratamiento se realizó, para esto se despliega una paleta de colores dando infinidad de combinaciones posibles para que no sea necesario repetir ningún color si no se quiere. Se realizó este prototipo adecuándolo en cada iteración hasta que fue aprobado por los usuarios y el tutor empresarial.

#### IV.3.4.3 Construcción

Una vez aprobado el prototipo de interfaz gráfica (Apéndice J), y definida la funcionalidad del módulo y los procesos que intervienen en la creación y consulta de tratamientos se procede a la construcción del módulo definido a continuación por el desarrollo de los siguientes procesos:

1. Crear, consultar, modificar y eliminar tratamiento.
2. Asignar color al tratamiento.
3. Seleccionar los dientes que podrán ser tratados con dicho tratamiento.
4. Asignar monto (costo).

#### IV.3.4.4 Implementación

En esta etapa se crearon diferentes tratamientos con diferentes características y costos, así como diferentes dientes asignados. Se comprobó que las validaciones funcionan correctamente, se validaron las reglas de negocio incluidas en el módulo, entre las que destacan no poder repetir tratamientos en el registro, es decir, que no se pudiera crear un tratamiento con igual nombre de uno existente.

### IV.3.5 Iteración 5: Diseñar e implementar módulo de enfermedades.

#### IV.3.5.1 Análisis y planificación de requerimientos

El análisis y planificación de requerimientos de este módulo se basó en el diagrama de caso de uso (Apéndice A.3) y en la información que se desea conocer de cada enfermedad, dichas enfermedades no tienen necesariamente una relación con la odontología pero pueden influir de manera indirecta al momento en el que el paciente recibe tratamiento, se determinó que las alergias también pueden influir de manera negativa en los tratamientos y por eso se tomó la decisión de incluirlas en el módulo. De cada enfermedad y/o alergia se conoce nombre, descripción y tipo.

#### IV.3.5.2 Diseño

Para el desarrollo de este módulo se diseñó un prototipo de interfaz gráfica (Apéndice I) que permitiera crear enfermedad, consultar, modificar toda su información y eliminarla, se diseñaron cuatro pantallas para realizar estos requerimientos, una en la que se listan todas las enfermedades y/o alergias que están registradas, una de consulta (Apéndice I.1), una de modificación (Apéndice I.2) en la que se puede ver en detalle una a una las enfermedades y/o alergias que se quiera y una para hacer eliminación (Apéndice I.3) si así se requiere. Se muestran en una tabla que las presenta en listas paginadas con 10 registros cada una.

#### IV.3.5.3 Construcción

Una vez aprobado el prototipo de interfaz gráfica (Apéndice I) y definida la funcionalidad del módulo y los procesos que intervienen en la creación y consulta de enfermedades y/o alergias se procede a la construcción del módulo definido a continuación por el desarrollo de los siguientes procesos:

1. Crear, consultar, modificar y eliminar Enfermedad y/o Alergia.
2. Agregar descripción.
3. Seleccionar el tipo.

#### IV.3.5.4 Implementación

En esta etapa se crearon diferentes enfermedades y alergias con diferentes características. Se comprobó que las validaciones funcionan correctamente, se validaron las reglas de negocio incluidas en el módulo, entre las que destacan no poder repetir enfermedades ni alergias en el registro, es decir, que no se pudiera crear una enfermedad ni alergia con igual nombre de una existente.

### IV.3.6 Iteración 6: Diseñar e implementar módulo de contratos.

#### IV.3.6.1 Análisis y planificación de requerimientos

El análisis y planificación de requerimientos de este módulo se basó en la información que se desea conocer de cada contrato y en el diagrama de caso de uso (Apéndice A.4), se maneja la fecha de inicio y la fecha de fin de cada contrato entre el Doctor y cada Aseguradora.

#### IV.3.6.2 Diseño

En esta etapa se diseñó el prototipo de interfaz gráfica (Apéndice G) que abarcara todos los requerimientos del módulo y permitirá a los doctores introducir el rango de fechas en las cuales tiene validez cada contrato, de esta forma el Doctor puede llevar control y no agendar citas ni realizar tratamientos a los asegurados cuya Aseguradora no tenga vigente el contrato.

#### IV.3.6.3 Construcción

Una vez aprobado el prototipo de interfaz gráfica (Apéndice G) y definida la funcionalidad del módulo y los procesos que intervienen en la creación y consulta de Contratos se procede a la construcción del módulo definido a continuación por el desarrollo de los siguientes procesos para cada Contrato:

1. Crear.
2. Modificar.
3. Eliminar.
4. Consultar.

### IV.3.6.4 Implementación

En esta etapa se crearon diferentes Contratos con diferentes características. Se comprobó que las validaciones funcionan correctamente, se validaron las reglas de negocio incluidas en el módulo, entre las que destacan no poder repetir contratos en el registro, es decir, que no se pudiera crear un contrato con una aseguradora si existe uno vigente.

### IV.3.7 Iteración 7: Diseñar e implementar la Interfaz de Integración y Comunicación con las Aseguradoras.

#### IV.3.7.1 Análisis y planificación de requerimientos

El análisis y planificación de requerimientos de este módulo se basó en las siguientes preguntas:

1. ¿Qué información se necesita intercambiar con las aseguradoras?

   La respuesta fue cobertura actualizada de cada paciente, facturas por cobrar y pagos recibidos.

2. ¿Qué se debe validar?

   Se debe validar que cada cita o tratamiento realizado al Paciente asegurado sea descontado del monto de su cobertura y se mantenga actualizada la información con la aseguradora, de esta manera evitamos realizar tratamientos o citas a pacientes cuya cobertura no cubra el costo de los mismos.

#### IV.3.7.2 Diseño

Se diseñará una interfaz la cual ejercerá la comunicación con las aseguradoras a través de un *web service* de tipo REST, se determina el uso de este tipo de servicio web porque soporta más lenguajes (XML, JSON, XHTML), esta comunicación será fundamental para la realización de los pagos de forma oportuna.

El servicio web servirá como puente entre los servicios, consumirá el API de las aseguradoras y a su vez servirá como API para ellas. Con esto el proceso de verificación de la información recibida por la Aseguradora se

realizará de manera más ágil puesto que toda la información recibida será digital y estará ordenada y agrupada por paciente.

Se busca que el doctor se beneficie al recibir información actualizada de la cobertura restante de cada paciente, ya que la misma estará sincronizada con las aseguradoras.

#### IV.3.7.3 Construcción

Se hace uso explícito de los métodos HTTP:

- Se utilizó *POST* para crear recursos en el servidor
- Se utilizó *GET* para obtener recursos
- Se utiliza *PUT* para cambiar el estado de un recurso o actualizarlo
- Se hace uso de *DELETE* para eliminar un recurso

#### IV.3.7.4 Implementación

En esta etapa se crearon diferentes conexiones como prueba de comunicación con las aseguradoras. Se comprobó que las validaciones funcionan correctamente, se validaron las reglas de negocio, entre las que destaca la cobertura del paciente, es decir, que no se pudiera realizar un tratamiento a un paciente cuya cobertura no cubra el valor del mismo.

## IV.3.8 Iteración 8: Diseñar e implementar módulo de pagos.

### IV.3.8.1 Análisis y planificación de requerimientos

El análisis y planificación de requerimientos de este módulo se basó en la información que se desea y se necesita conocer de los pagos pendientes por parte de los pacientes o las aseguradoras así como en el diagrama de caso de uso (Apéndice A.8).

De los pagos realizados por el paciente se conocerá el tipo de pago, es decir, si es realizado en efectivo, tarjeta débito, tarjeta de crédito, transferencia o cheque, guardando en cada caso la información relevante como número de cheque o el número de confirmación de la transferencia.

Si el paciente indica que el pago será cargado a la aseguradora sólo debe escoger a cuál de ellas en caso que posea más de un seguro, y ésta información de cobro será enviada a la aseguradora respectiva a través de *Web Service*, y por la misma vía la aseguradora dará respuesta si acepta o no el cobro.

### IV.3.8.2 Diseño

Para el desarrollo de este módulo se diseñó un prototipo de interfaz gráfica (Apéndice O) que permitiera llevar control de los pagos, consultar, registrar los pagos realizados directamente por el paciente y enviar toda la información a la aseguradora que sea responsable del pago, se diseñaron dos pantallas para realizar estos requerimientos, una pantalla (Apéndice O) en la que se listan todos las citas que tienen tratamientos sin cancelar, en la ésta se podrá registrar quien hará el pago, en caso de ser la aseguradora se registrará cuál de ellas y se enviará la información y en caso que sea un pago directo del

paciente se mostrará una nueva pantalla (Apéndice O.1) en la que se podrá seleccionar el tipo de pago y la información relevante del mismo, en ésta pantalla se lleva control si realizó el pago total o pago parcial de la deuda.

#### IV.3.8.3 Construcción

Una vez aprobado el prototipo de interfaz gráfica, y definida la funcionalidad del módulo y los procesos que intervienen en la consulta y registro de pagos se procede a la construcción del módulo definido a continuación por el desarrollo de los siguientes procesos:

1. Consultar citas con tratamientos pendientes por pagar.
2. Asignar responsable del pago (paciente o aseguradora).
3. Para pago por paciente registrar la forma de pago.
4. Para pago por aseguradora hacer registro y enviar la información para el cobro.

#### IV.3.8.4 Implementación

En esta etapa se cargaron diferentes pagos por paciente y por aseguradora. Se comprobó que las validaciones funcionan correctamente, se validaron las reglas de negocio incluidas en el módulo y se validó que la información era enviada a la aseguradora en los casos que fuera necesario.

### IV.3.9 Iteración 9: Diseñar e implementar módulo de aseguradoras.

#### IV.3.9.1 Análisis y planificación de requerimientos

El análisis y planificación de requerimientos de este módulo se basó en la información que se desea conocer de cada aseguradora y en el diagrama de caso de uso (Apéndice A.7), dichas aseguradoras al estar cargadas en el Sistema podrán ser seleccionadas al momento de registrar un nuevo paciente que este asegurado con alguna de ellas o cuando uno de los pacientes existentes decida registrarse como asegurado. De cada aseguradora se conoce nombre, descripción y estatus.

#### IV.3.9.2 Diseño

Para el desarrollo de este módulo se diseñó un prototipo de interfaz gráfica (Apéndice K) que permitiera crear aseguradora, consultar, modificar toda su información y eliminarla, se diseñaron cuatro pantallas para realizar estos requerimientos, una en la que se listan todas las aseguradoras que están registradas, crear aseguradora (Apéndice K.1), una de modificación (Apéndice K.2) en la que se puede ver en detalle una a una las aseguradoras que se quiera y una para hacer eliminación (Apéndice K.3) si así se requiere. Se muestran en una tabla que las presenta en listas paginadas con 10 registros cada una, esto para facilitar al usuario la navegación en el sistema.

#### IV.3.9.3 Construcción

Una vez aprobado el prototipo de interfaz gráfica y definida la funcionalidad del módulo y los procesos que intervienen en la creación y consulta de aseguradoras se procede a la construcción del módulo definido a

continuación por el desarrollo de los siguientes procesos:

1. Crear, consultar, modificar y eliminar aseguradora.
2. Agregar descripción.
3. Seleccionar el estatus.

#### IV.3.9.4 Implementación

En esta etapa se crearon diferentes aseguradoras. Se comprobó que las validaciones funcionan correctamente, se validaron las reglas de negocio incluidas en el módulo, entre las que destacan no poder repetir aseguradoras en el registro, es decir, que no se pudiera crear una aseguradora con igual nombre de una existente.

### IV.3.10 Iteración 10: Diseñar e implementar módulo de citas.

#### IV.3.10.1 Análisis y planificación de requerimientos

El análisis y planificación de requerimientos de este módulo se basó en el diagrama de caso de uso (Apéndice A.5) y en la necesidad de llevar el control de citas diario y relacionar los tratamientos de cada paciente a la cita en la que fueron realizados, estas citas cumplen un rol importante en el módulo de pagos puesto que en cada una de ellas se ha realizado un tratamiento y al seleccionar la cita se puede ver cuánto debe el paciente. De cada cita se conoce el doctor, nombre del paciente, estatus y la fecha de la misma.

#### IV.3.10.2 Diseño

Para el desarrollo de este módulo se diseñó un prototipo de interfaz gráfica (Apéndice H) que permitiera crear cita, consultar, modificarla y eliminarla, se diseñaron cinco pantallas para realizar estos requerimientos, una en la que se listan todas las citas que están registradas, una de crear cita (Apéndice H.1), una de modificación (Apéndice H.2) en la que se puede ver en detalle una a una las citas y modificar la que se quiera, una pantalla (Apéndice H.4 – H.5) en la que se podrá registrar en el gráfico dental que tratamiento se realizó en esa cita y una para hacer eliminación (Apéndice H.3) si así se requiere. Se muestran en una tabla que las presenta en listas paginadas con 10 registros cada una, esto para hacer amigable al usuario la navegación en el sistema.

#### IV.3.10.3 Construcción

Una vez aprobado el prototipo de interfaz gráfica (Apéndice H) y definida la funcionalidad del módulo y los procesos que intervienen en la creación y consulta de citas se procede a la construcción del módulo definido a continuación por el desarrollo de los siguientes procesos:

1. Crear, consultar, modificar y eliminar cita.
2. Seleccionar el estatus.
3. Agregar detalle del tratamiento realizado en el gráfico dental.

#### IV.3.10.4 Implementación

En esta etapa se crearon diferentes citas. Se comprobó que las validaciones funcionan correctamente, se validaron las reglas de negocio incluidas en el módulo, entre las que destacan el detalle del tratamiento realizado en la misma en el gráfico dental.

### IV.3.11 Iteración 11: Diseñar e implementar módulo de reportes.

#### IV.3.11.1 Análisis y planificación de requerimientos

En reunión con el tutor empresarial y algunos usuarios se realizó el análisis y planificación de requerimientos de este módulo, se determinó cuáles son los reportes más relevantes, siendo ocho los reportes generados finalmente por el Sistema:

- Citas por rango de fechas
- Tratamientos por paciente
- Pacientes por aseguradora
- Pagos realizados por tratamiento
- Contratos por paciente
- Contratos por aseguradora
- Pagos realizados por rango de fechas
- Enfermedades más frecuentes

#### IV.3.11.2 Diseño

Basado en el análisis de los requerimientos, se diseñó la interfaz gráfica que consta de una sola pantalla (Apéndice P) en la que se puede seleccionar cual es el reporte que se quiere, según el elegido la pantalla dinámicamente pedirá la información necesaria para generarlo ya que para cada uno se necesita información diferente; para el reporte de citas por rango de fechas en la pantalla se muestran dos cuadros en los que se podrá señalar la fecha de inicio y fecha de fin de la información que queremos; en tratamientos por paciente se pide seleccionar el paciente del que se quiere información; en

pacientes por aseguradora se pide seleccionar de cuál de ellas se quiere reporte; pagos realizados por tratamiento se pide el tratamiento; en contratos por paciente se pide el nombre del mismo; del mismo modo en contratos por aseguradora se pide el nombre de la misma; se pide el rango de fechas para mostrar el reporte de pagos realizados y para el de enfermedades más frecuentes solo es necesario dar a cargar reporte.

#### IV.3.11.3 Construcción

Una vez aprobado el prototipo de interfaz gráfica (Apéndice M) y definida la funcionalidad del módulo y los procesos que intervienen en la generación de reportes se procede a la construcción del módulo que consta de la lista de posibles reportes, el espacio en el que se pide la información necesaria para generarlo y el botón de cargar, una vez cargado se ve en la parte inferior de la pantalla.

#### IV.3.11.4 Implementación

Se generaron todos los reportes con diferentes características; para el reporte de citas por rango de fechas se ingresaron diferentes rangos y se validó directo en la base de datos si la información mostrada era correcta; se generaron reportes de enfermedades frecuentes, de contratos por paciente, etc. En todos los casos se validó directo en base de datos si la información mostrada era correcta obteniendo resultados satisfactorios.

### IV.3.12 Iteración 12: Módulo de Usuarios y Roles (Administración).

#### IV.3.12.1 Análisis y planificación de requerimientos

El análisis y planificación de requerimientos de este módulo se basó en la necesidad de administrar el sistema, crear módulos, usuarios y roles.

Aunque el crear nuevos módulos no es parte del alcance de este proyecto, en reunión con el tutor empresarial se tomó la decisión de dar esta opción al usuario como un plus en la configuración del Sistema.

El manejo de usuarios y roles es indispensable en cualquier Sistema para controlar la interacción de los usuarios y las acciones que puedan desarrollar en el mismo, se necesitan escalar los permisos para el resguardo del Sistema.

#### IV.3.12.2 Diseño

Se diseñó un prototipo de interfaz gráfica para el módulo (Apéndice R) con tres pantallas similares entre sí, una para módulos, otra para usuarios y finalmente una para roles.

#### IV.3.12.3 Construcción

Para la construcción de este módulo, se construyó la interfaz gráfica del módulo, compuesta por la pantalla de crear módulo, la pantalla de crear usuario y la de crear rol.

#### IV.3.12.4 Implementación

En esta etapa se realizó una prueba del proceso de configuración de usuarios, se le asignó el rol de director a un usuario prueba1 para que pudiera realizar todos los cambios y ajustes en el sistema y a otro usuario prueba2 al

que se asignó el rol de secretaria para que solo pudiera acceder a ciertas funcionalidades del Sistema como las citas, se comprobó el correcto funcionamiento de la administración de usuarios y roles.

# CAPÍTULO V: RESULTADOS

## Aporte Funcional

- **Rediseñar Proceso de Gestión de Pagos**

Después de realizar evaluaciones en las que estuvieron involucrados los usuarios finales (Secretarias, Doctores y Directores) se determinó que al proceso de gestión de pagos no podía cambiar lo suficiente para justificar un rediseño del mismo, por lo tanto se realizó una automatización del proceso actual, ahora permite a los doctores crear solicitudes de pago a las aseguradoras a través del sistema, con el proceso existente las solicitudes se realizan de forma manual, el envío que realizan las instituciones o doctores de dichas solicitudes es tedioso para las dos partes (doctores y aseguradoras) puesto que implica el manejo de papelería y el transporte de la misma, adicionalmente los analistas de las aseguradoras deben cargar manualmente toda esa información en su sistema para así poder empezar el proceso de validación de la data y proceder a autorizar o no los pagos.

Las aseguradoras se ven beneficiadas con la implementación de esta automatización porque ayuda a evitar posibles fraudes por cobros indebidos en tratamientos o casos en los que se pretenda cobrar varias veces el mismo trabajo realizado, de igual modo se evita la transcripción manual por parte de los analistas.

Los doctores son los responsables de autogestionar sus cobros a las aseguradoras y cargarlos al sistema, y los analistas de las aseguradoras sólo se dedican a analizar cada solicitud de pago perteneciente a cada doctor y determinar si se aprueba o no su pago.

En el diseño del proceso de autogestión se puede observar un cambio en el proceso de creación de solicitudes de pago, donde ya el doctor asociado o el

encargado del proceso en la institución no tienen la necesidad de asistir a las instalaciones de la aseguradora para que llevar las solicitudes de pago.

- **Rediseñar Proceso de Control de Tratamientos**

Se rediseñó el proceso de control de tratamientos que permite a los doctores o sus asistentes cargar en el sistema en tiempo real el tratamiento que se está realizando al paciente, del mismo modo permite a las aseguradoras llevar un mejor control de dichos procedimientos ya que se llevaba de forma manual, a la aseguradoras llegaban los formatos de los tratamientos por cada paciente de forma física y los analistas debían registrar en su sistema estos tratamientos, sin embargo en conversaciones con los usuarios se evidenció que en muchos casos se podía repetir la carga de un mismo tratamiento en los sistemas de las aseguradoras y generar cobros múltiples para un mismo procedimiento en un mismo paciente, con el rediseño las aseguradoras tienen digitalizados todos los tratamientos y cuentan con la nueva funcionalidad del historial del paciente de forma gráfica gracias al uso del gráfico dental (Apéndice Q).

**Aporte Tecnológico**

- **Evaluar la plataforma tecnológica (lenguaje de programación, manejador de base de datos, framework de desarrollo) y seleccionar la más adecuada para el desarrollo del aplicativo web:** se evaluaron las herramientas de desarrollo IDE Microsoft Visual Studio y SharpDevelop (Tabla 2), resultando IDE Microsoft Visual Studio la herramienta más adecuada para el desarrollo del sistema.

En relación al cuadro comparativo (Tabla 3) de las herramientas de desarrollo de base de datos se aprecia que todas cumplen con las características mínimas necesarias para el desarrollo del sistema pero es Navicat la elegida por ofrecer más características entre las que destacan:

- ✓ Analizar, repasar y reparar tablas
- ✓ Administrador Visual para la gestión de usuarios y sus privilegios
- ✓ Programar copias de seguridad
- ✓ Importar y Exportar datos desde MS Excel y MS Access
- ✓ Copia de seguridad de bases de datos y scripts de SQL
- ✓ Sincronización de Datos y Estructuras
- ✓ Modelado de datos
- ✓ Vista de Diagrama ER

En cuanto al cuadro comparativo (Tabla 4) de los Framework se aprecia que los dos cumplen con las características necesarias para el desarrollo del sistema pero es ASP .NET MVC el elegido por ofrecer mejores características como la separación clara de intereses y el rendimiento más rápido.

La evaluación de las herramientas se puede observar en el Capítulo IV -

Investigación Preliminar.

- **Diseñar e implementar una interfaz de integración y comunicación con las aseguradoras (al menos con una de ellas):** Se diseñó una interfaz la cual ejerce la comunicación con las aseguradoras a través de un web service, esta comunicación es fundamental para la realización de los pagos de forma oportuna.

  Con esto el proceso de verificación de la información recibida por la aseguradora se realizará de manera más ágil puesto que toda la información recibida será digital y estará ordenada y agrupada por paciente.

  Se busca que el doctor se beneficie al recibir información actualizada de la cobertura restante de cada paciente, ya que la misma estará sincronizada con las aseguradoras. El desarrollo de este mecanismo de seguridad se puede observar en el Capítulo IV - Iteración 7.

A continuación se detallan los resultados que se obtuvieron con base en los objetivos específicos propuestos. Cabe destacar que el sistema cumplió en su totalidad con cada uno de los objetivos y funcionalidades:

1. **Diseñar e implementar la base de datos:** se obtuvo como resultado el modelo Entidad-Relación de la base de datos. Este modelo se implementó e integró completamente y se basa en las necesidades del cliente. La base de datos permite guardar y manipular los datos enviados desde el sistema, contiene las tablas necesarias para el funcionamiento de cada módulo del sistema.

2. **Diseñar e Implementar el módulo de pacientes:** se obtuvo como resultado un módulo que es capaz de crear pacientes, consultar, modificar toda su información y eliminar si se requiere. Se diseñaron cuatro pantallas para realizar estos requerimientos, la pantalla de nuevo paciente, en la que se asigna nombres, apellidos, información de contacto, también se puede asociar diferentes enfermedades y/o alergias así como los tratamientos que tiene con anterioridad; una en la que se listan todas los pacientes que están registrados, listados en páginas de 10 registros para hacer amigable al usuario la navegación por el sistema; una pantalla de consulta y modificación en la que se puede ver en detalle uno a uno los pacientes que se quiera y hacer las respectivas modificaciones y registro de nuevos tratamientos realizados; y una para hacer eliminación si así se requiere. El desarrollo de este módulo se puede observar en el Capítulo IV - Iteración **2**.

3. **Diseñar e implementar el módulo de doctores:** se obtuvo un módulo que permite crear doctor, consultar, modificar toda su información y eliminar si se requiere, se diseñaron cuatro pantallas para realizar estos requerimientos, la pantalla de nuevo doctor, en la que se asigna nombres, apellidos, información de contacto; una en la que se listan todas los doctores que están registrados, listados en páginas de 10 registros para hacer amigable al usuario la navegación por el sistema; una pantalla de consulta y modificación en la que se puede ver en detalle uno a uno los doctores que se quiera y modificar lo que sea necesario; y una para hacer eliminación si así se requiere. En este módulo también se puede asociar las aseguradoras con las que tiene contrato. El desarrollo de este módulo se puede observar en el Capítulo IV - Iteración **3**.

4. **Diseñar e implementar el módulo de tratamientos**: se obtuvo un módulo que permite crear tratamiento, consultar, modificar toda su información y eliminar si se requiere, se diseñaron cuatro pantallas para realizar estos requerimientos, la pantalla de nuevo tratamientos, en la que se asigna nombre, descripción, color (este color será el que identifique al tratamiento en el gráfico dental), monto (costo) y la lista de los dientes que pueden ser intervenidos con el mismo; una en la que se listan todas los tratamientos que están registrados, listados en páginas de 10 registros para hacer amigable al usuario la navegación por el sistema; una pantalla de consulta en la que se puede ver en detalle uno a uno los tratamientos que se quiera y una para hacer eliminación si así se requiere.

En este módulo también se puede asociar los tratamientos para cada diente o para todos si así se requiere, se asigna un color con el que se podrá identificar fácilmente en el gráfico dental que tratamiento se realizó, para esto se despliega una paleta de colores dando infinidad de combinaciones posibles para que no sea necesario repetir ningún color si no se quiere. El desarrollo de este módulo se puede observar en Capítulo IV - Iteración 4.

5. **Diseñar e implementar el módulo de enfermedades**: se obtuvo un módulo que permite crear enfermedad, consultar, modificar toda su información y eliminarla, se diseñaron cuatro pantallas para realizar estos requerimientos, una en la que se listan todas las enfermedades y/o alergias que están registradas, una de consulta y modificación en la que se puede ver en detalle una a una las enfermedades y/o alergias que se quiera y una para hacer eliminación si así se requiere. Se muestran en una tabla que las presenta en listas paginadas con 10 registros cada una, esto para hacer amigable al usuario la navegación en el sistema. El desarrollo de este módulo se

puede observar en el Capítulo IV - Iteración **5.**

6. **Diseñar e implementar el módulo de contratos:** se obtuvo un módulo que abarca todos los requerimientos del mismo y permite a los doctores introducir el rango de fechas en las cuales tiene validez cada contrato, de esta forma el Doctor puede llevar control y no agendar citas ni realizar tratamientos a los asegurados cuya Aseguradora no tenga vigente el contrato. El desarrollo de este módulo se puede observar en el Capítulo IV - Iteración **6.**

**7. Diseñar e implementar el módulo de pagos:** se obtuvo un módulo que permite llevar control de los pagos, consultar, registrar los pagos realizados directamente por el paciente y enviar toda la información a la aseguradora que sea responsable del pago, se diseñaron dos pantallas para realizar estos requerimientos, una pantalla en la que se listan todos las citas que tienen tratamientos sin cancelar, en ésta se podrá registrar quien hará el pago, en caso de ser la aseguradora se registrará cuál de ellas y se enviará la información y en caso que se a un pago directo del paciente se mostrará una nueva pantalla en la que se podrá seleccionar el tipo de pago y la información relevante del mismo, en ésta pantalla se lleva control si realizó el pago total o pago parcial de la deuda. El desarrollo de este mecanismo de seguridad se puede observar en el Capítulo IV - Iteración **8.**

**8. Diseñar e implementar el módulo de aseguradoras:** se obtuvo un módulo permite crear aseguradora, consultar, modificar toda su información y eliminarla, se diseñaron cuatro pantallas para realizar estos requerimientos, una en la que se listan todas las aseguradoras que están registradas, una de consulta y modificación en la que se puede ver en detalle una a una las aseguradoras que se quiera y una para hacer eliminación si así se requiere. Se muestran en una tabla que las presenta en listas

paginadas con 10 registros cada una, esto para hacer amigable al usuario la navegación en el sistema. El desarrollo de este mecanismo de seguridad se puede observar en el Capítulo IV - Iteración 9.

**9. Diseñar e implementar el módulo de citas:** se obtuvo un módulo permite crear cita, consultar, modificarla y eliminarla, se diseñaron cinco pantallas para realizar estos requerimientos, una en la que se listan todas las citas que están registradas, una de consulta y modificación en la que se puede ver en detalle una a una las citas que se quiera, una pantalla en la que se podrá registrar en el gráfico dental que tratamiento se realizó en esa cita y una para hacer eliminación si así se requiere. Se muestran en una tabla que las presenta en listas paginadas con 10 registros cada una, esto para hacer amigable al usuario la navegación en el sistema. El desarrollo de este mecanismo de seguridad se puede observar en el Capítulo IV - Iteración **10.**

**10. Diseñar e implementar el módulo de reportes:** se diseñó la interfaz gráfica que consta de una sola pantalla en la que se puede seleccionar cual es el reporte que se quiere, según el elegido la pantalla dinámicamente pedirá la información necesaria para generarlo ya que para cada uno se necesita información diferente; para el reporte de citas por rango de fechas en la pantalla se muestran dos cuadros en los que se podrá señalar la fecha de inicio y fecha de fin de la información que queremos; en tratamientos por paciente se pide seleccionar el paciente del que se quiere información; en pacientes por aseguradora se pide seleccionar de cuál de ellas se quiere reporte; pagos realizados por tratamiento se pide el tratamiento; en contratos por paciente se pide el nombre del mismo; del mismo modo en contratos por aseguradora se pide el nombre de la misma; se pide el rango de fechas para mostrar el reporte de pagos realizados y para el de enfermedades más frecuentes solo es

necesario dar a cargar reporte. El desarrollo de este mecanismo de seguridad se puede observar en el Capítulo IV - Iteración **11**.

**11. Diseñar e implementar el módulo de usuarios y roles:** se obtuvo un módulo el cual permite configurar usuarios. Adicionalmente, permite asignarle un nivel de aprobación a cada usuario configurado. El desarrollo de este módulo se puede observar en el Capítulo IV - Iteración 12.

# CAPITULO VI: CONCLUSIONES Y RECOMENDACIONES

## VI.1 CONCLUSIONES

Durante el desarrollo de este proyecto se pudo analizar la complejidad del proceso de pago de servicios pertenecientes a pacientes afiliados a seguradoras, comenzando desde la solicitud de pago, pasando por la carga por parte de los analistas de las aseguradoras y el análisis de la información del servicio prestado a sus asegurados, finalizando con la aprobación o negación de pago perteneciente al servicio.

El sistema permitirá llevar la administración total del consultorio o institución a la que pertenezca el doctor o doctores, podrán tener control de citas, llevar el historial de tratamientos realizados al paciente de manera gráfica y detallada, y el aplicativo permitirá llevar control de los pagos o abonos realizados por concepto de servicios y/o tratamientos prestados bien sea por parte del paciente o bien por parte de la aseguradora.

En cuanto al pago por parte de las aseguradoras se evitará el envío de solicitudes físicas de pago por parte de los doctores o las instituciones ya que dichas solicitudes se realizarán en línea y por esta misma vía las aseguradoras enviarán respuesta de las mismas, con esta interacción en línea podrán llevar control también del saldo restante de la cobertura correspondiente a cada asegurado evitando fraudes y posibles cobros injustificados.

## VI.2 RECOMENDACIONES

Se recomienda desarrollar e integrar mayor seguridad a la data correspondiente a la información médica de los pacientes ya que este tipo de sistemas pertenecientes al área Salud manejan información confidencial que puede ser utilizada de manera inescrupulosa por personas ajenas a la institución o consultorio odontológico, la parte de seguridad de la información no estuvo comprendida en el desarrollo de este proyecto. De este modo se garantizará a los pacientes que toda la información concerniente a su estado de salud e información general de carácter privado no podrá ser manipulada.

Adicionalmente, se recomienda integrar el sistema con un módulo de facturación. Actualmente se mantiene control de los pagos realizados, pero no se genera la factura de los mismos directamente en el sistema.

Como recomendación final, estandarizar una API para todas las aseguradoras para de esta manera unificar las conexiones y no tener la necesidad de realizar configuraciones independientes para cada compañía de seguros.

## BIBLIOGRAFÍA

[1] ¿Qué es .NET Framework? Disponible en: https://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx [Consultada en Junio 2016].

[2] Definición ABC. Definición de Aseguradora. Disponible en: http://www.definicionabc.com/derecho/aseguradora.php [Consultada en: Julio 2016].

[3] Información General sobre ASP.NET. Disponible en: https://msdn.microsoft.com/es-es/library/4w3ex9c2(v=vs.100).aspx [Consultada en Julio 2016].

[4] Información general sobre ASP.NET MVC. Disponible en: https://msdn.microsoft.com/es-es/library/dd381412(v=vs.108).aspx [Consultada en Agosto 2016].

[5] Definición ABC. Definición de Base de datos. Disponible en: http://www.definicionabc.com/base-de-datos.php [Consultada en: Julio 2016].

[6] ¿Qué es Bootstrap? Disponible en: https://alanchavez.com/si-eres-desarrollador-web-debes-utilizar-bootstrap-y-punto/ [Consultada en Julio 2016].

[7] C#. Disponible en: https://msdn.microsoft.com/es-es/library/kx37x362.aspx [Consultada en Julio 2016].

[8] Glosario de Términos. Cobertura. Disponible en: http://www.sudeaseg.gob.ve/?post_type=document&p=2774 [Consultada en: Junio 2016].

[9] Definición de Contrato. Disponible en: http://definicion.de/contrato/ [Consultada en: Junio 2016].

[10] Control de Versiones. Disponible en: https://git-scm.com/book/es/v1/Acerca-del-control-de-versiones [Consultada en: Julio 2016].

[11] Entity Framework. Disponible en: https://msdn.microsoft.com/es-es/data/ef.aspx [Consultada en Agosto 2016].

[12] ¿Qué es un Framework? Disponible en: http://edgargomez.es/que-es-un-framework/#.Vk6ZLNIrIrh [Consultada en Junio 2016].

[13] IDE de Programación. Disponible en: http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n [Consultada en Junio 2016].

[14] Metodología RAD. Disponible en: https://curiosisimos.files.wordpress.com/2009/12/modelo-de-desarrollo-rapido-de-aplicaciones.pdf [Consultada en Julio 2016].

[15] Microsoft SQL Server. Disponible en: https://msdn.microsoft.com/es-es/library/bb545450.aspx [Consultada en: Julio 2016].

[16] Microsoft Visual Studio. Disponible en: https://msdn.microsoft.com/es-es/library/fx6bk1f4(v=vs.100).aspx [Consultada en Julio 2016].

[17] Ecured. Definición de Navicat. Disponible en: http://www.ecured.cu/index.php/Navicat [Consultada en: Julio 2016].

[18] ¿Qué es REST? Disponible en: http://asiermarques.com/2013/conceptos-sobre-apis-rest/ [Consultada en Agosto 2016].

[19] Definición ABC. Definición de seguro. Disponible en: http://www.definicionabc.com/derecho/seguro.php [Consultada en: Julio 2016].

[20] Definición de Servicio Web. Disponible en: http://www.ehu.eus/mrodriguez/archivos/csharppdf/ServiciosWeb/WebServices.pdf [Consultada en Julio 2016].

[21] SharpDevelop. Disponible en: https://es.wikipedia.org/wiki/SharpDevelop [Consultada en Julio 2016].

[22] Definición de SOAP. Disponible en: http://www.alegsa.com.ar/Dic/soap.php [Consultada en Julio 2016].

[23] Definición de SQL. Disponible en: http://proyectobasesdedatos.galeon.com/enlaces1416754.html [Consultada en Junio 2016].

[24] Definición ABC. Definición de Tratamiento. Disponible en: http://www.definicionabc.com/salud/tratamiento.php [Consultada en: Junio 2016].

## GLOSARIO

**Aseguradora:** Una aseguradora [2], también designada en algunas partes como compañía de seguros, puede ser una persona física o una empresa que se ocupa de la aseguración de riesgos a terceros, es decir, su principal actividad consiste en resguardar a aquellos bienes materiales de determinados riesgos a los que están expuestos.

**Cobertura:** una cobertura [8] es el riesgo que cubre la póliza contratada con todas sus limitaciones delimitaciones y exclusiones.

**Contrato:** un contrato [9] es un término con origen en el vocablo latino contractus que nombra al convenio o pacto, ya sea oral o escrito, entre partes que aceptan ciertas obligaciones y derechos sobre una materia determinada. El documento que refleja las condiciones de este acuerdo también recibe el nombre de contrato.

**Seguro:** un seguro [19] es un contrato que se establece entre dos partes y que tiene como objetivo asegurar determinado bien o elemento.

# APÉNDICE

## APÉNDICE A. DIAGRAMAS DE CASOS DE USO.

## APÉNDICE A.1. DIAGRAMA DE CASO DE USO MODULO PACIENTES.

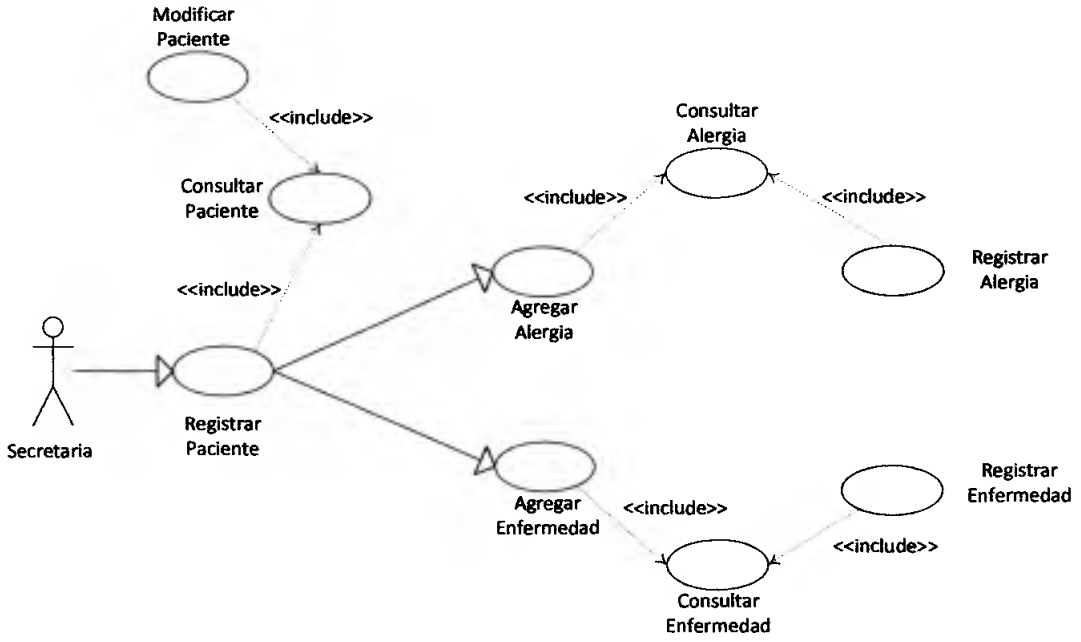



Figura 2: Caso de Uso Módulo Pacientes
Fuente: Elaboración Propia

## APÉNDICE A.2. DIAGRAMA DE CASO DE USO MODULO DOCTORES.

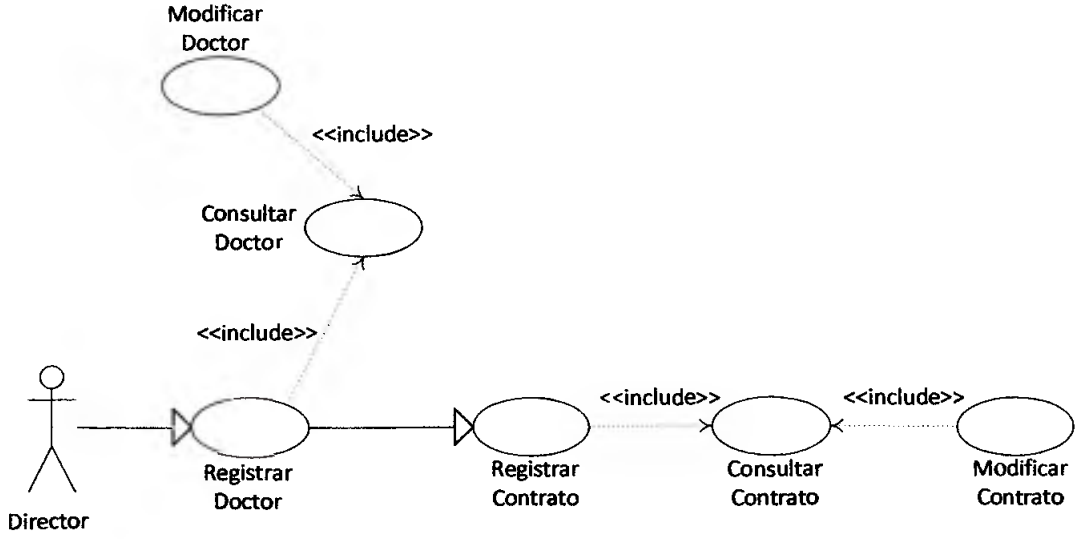



Figura 3: Caso de Uso Módulo Doctores
Fuente: Elaboración Propia

## APÉNDICE A.3. DIAGRAMA DE CASO DE USO MODULO ENFERMEDADES.

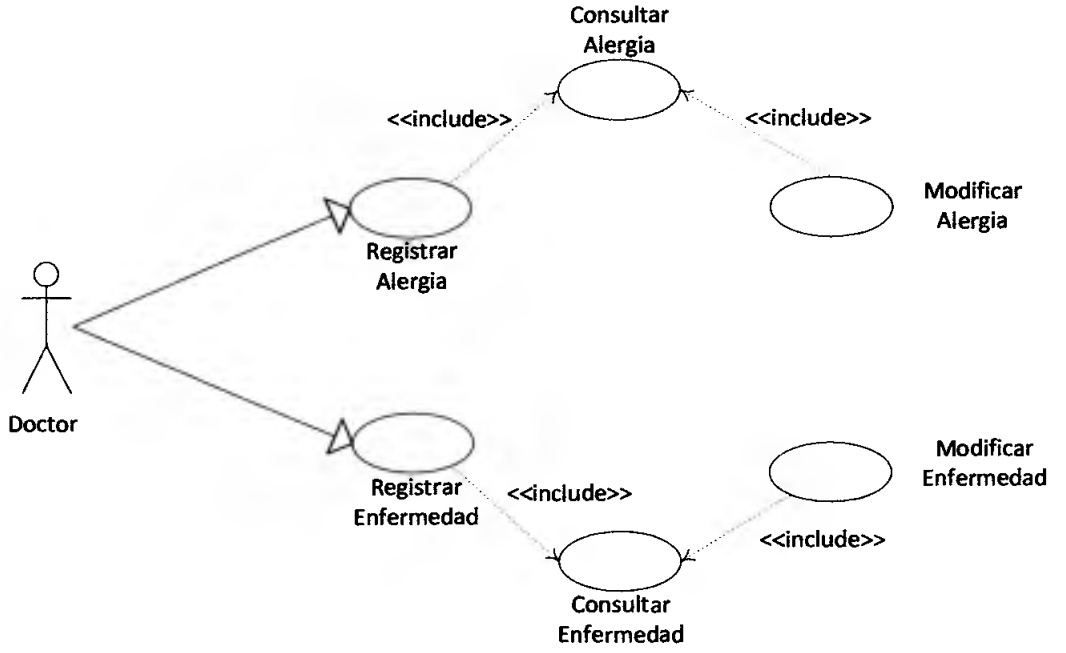



Figura 4: Caso de Uso Módulo Enfermedades
Fuente: Elaboración Propia

## APÉNDICE A.4. DIAGRAMA DE CASO DE USO MODULO CONTRATOS.

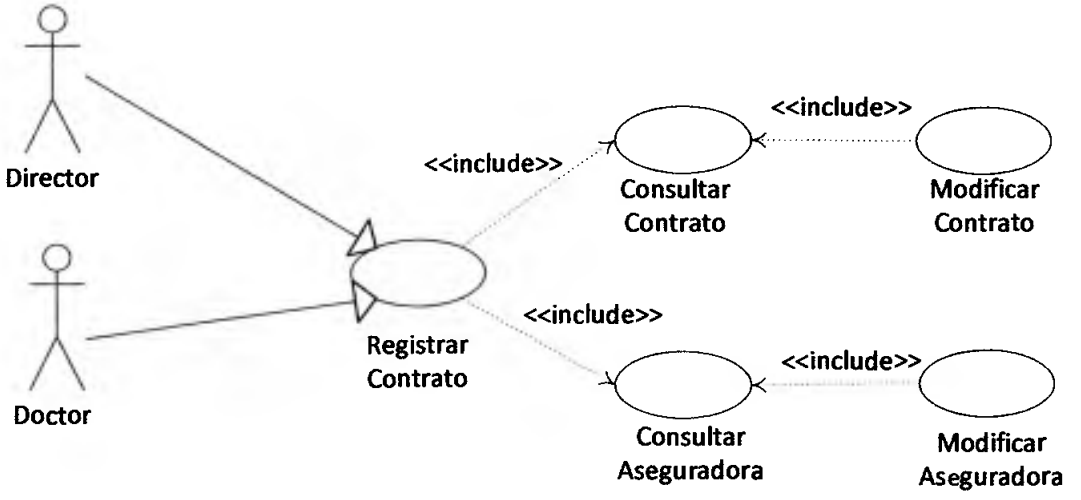



Figura 5: Caso de Uso Módulo Contratos
Fuente: Elaboración Propia

## APÉNDICE A.5. DIAGRAMA DE CASO DE USO MODULO CITAS.

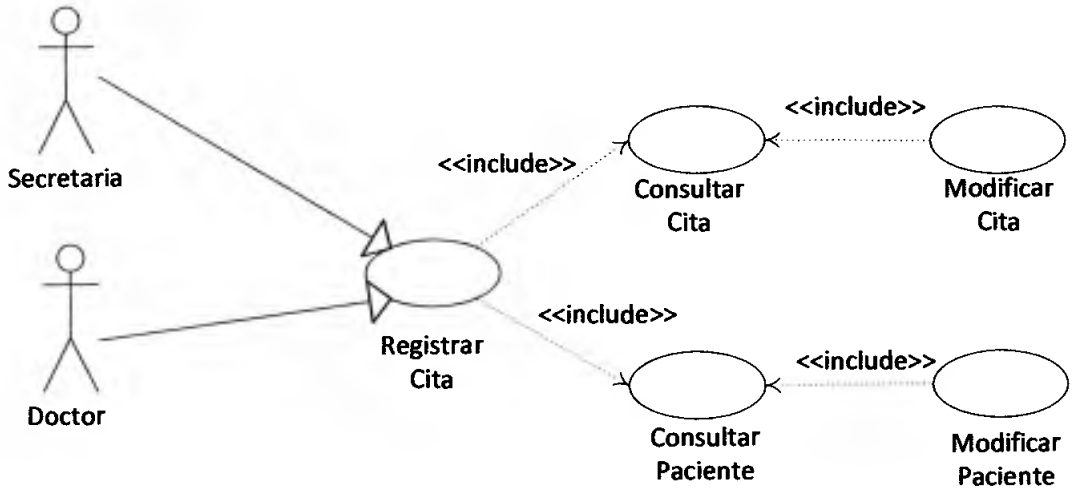



Figura 6: Caso de Uso Módulo Citas
Fuente: Elaboración Propia

## APÉNDICE A.6. DIAGRAMA DE CASO DE USO MODULO TRATAMIENTOS.

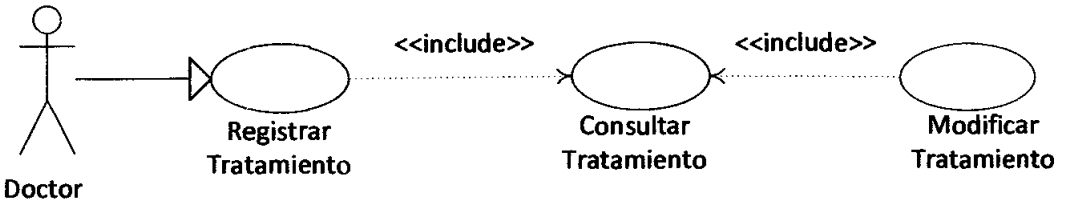



Figura 7: Caso de Uso Módulo Tratamientos
Fuente: Elaboración Propia

## APÉNDICE A.7. DIAGRAMA DE CASO DE USO MODULO ASEGURADORAS.

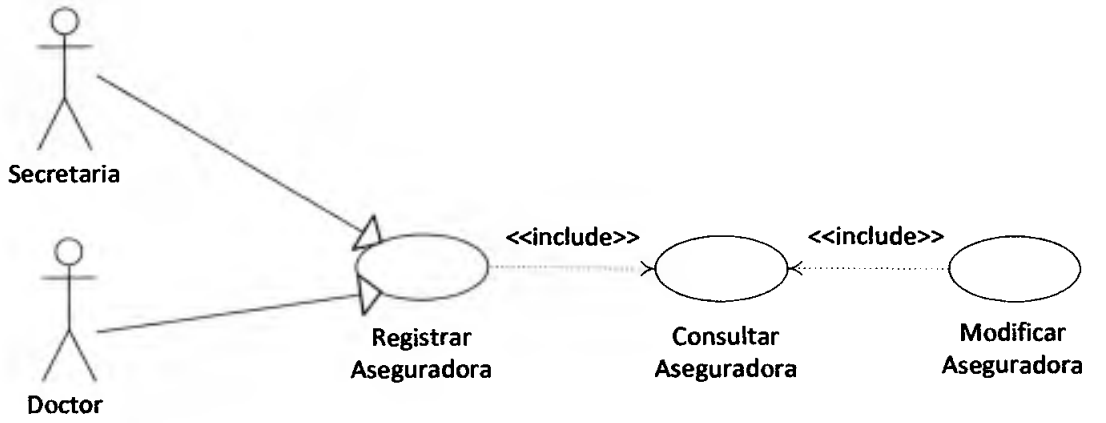



Figura 8: Caso de Uso Módulo Aseguradoras
Fuente: Elaboración Propia

## APÉNDICE A.8. DIAGRAMA DE CASO DE USO MODULO PAGOS.

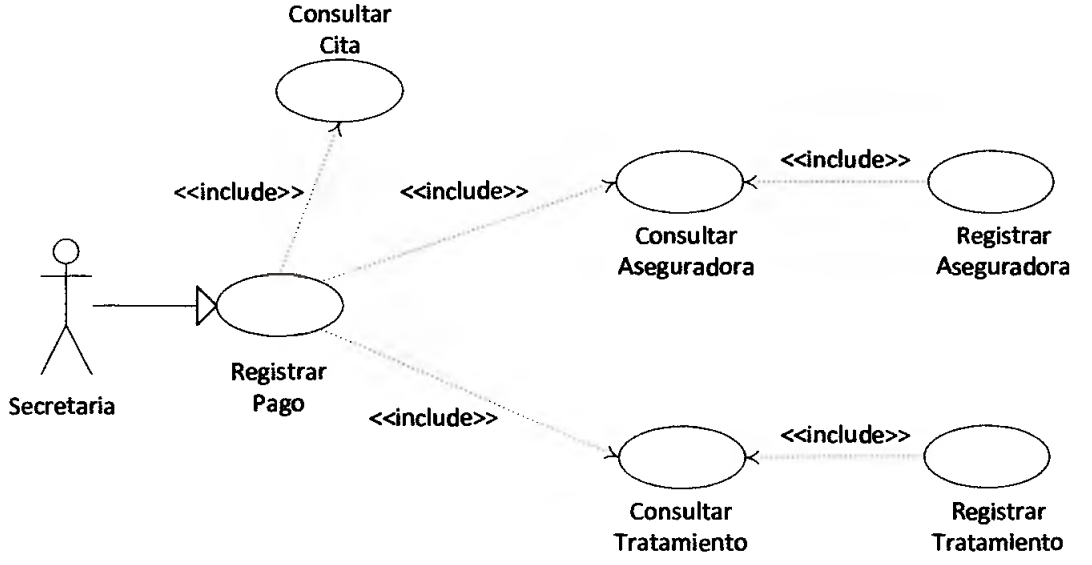



Figura 9: Caso de Uso Módulo Pagos
Fuente: Elaboración Propia

# APÉNDICE B. MODELO ENTIDAD RELACIÓN.

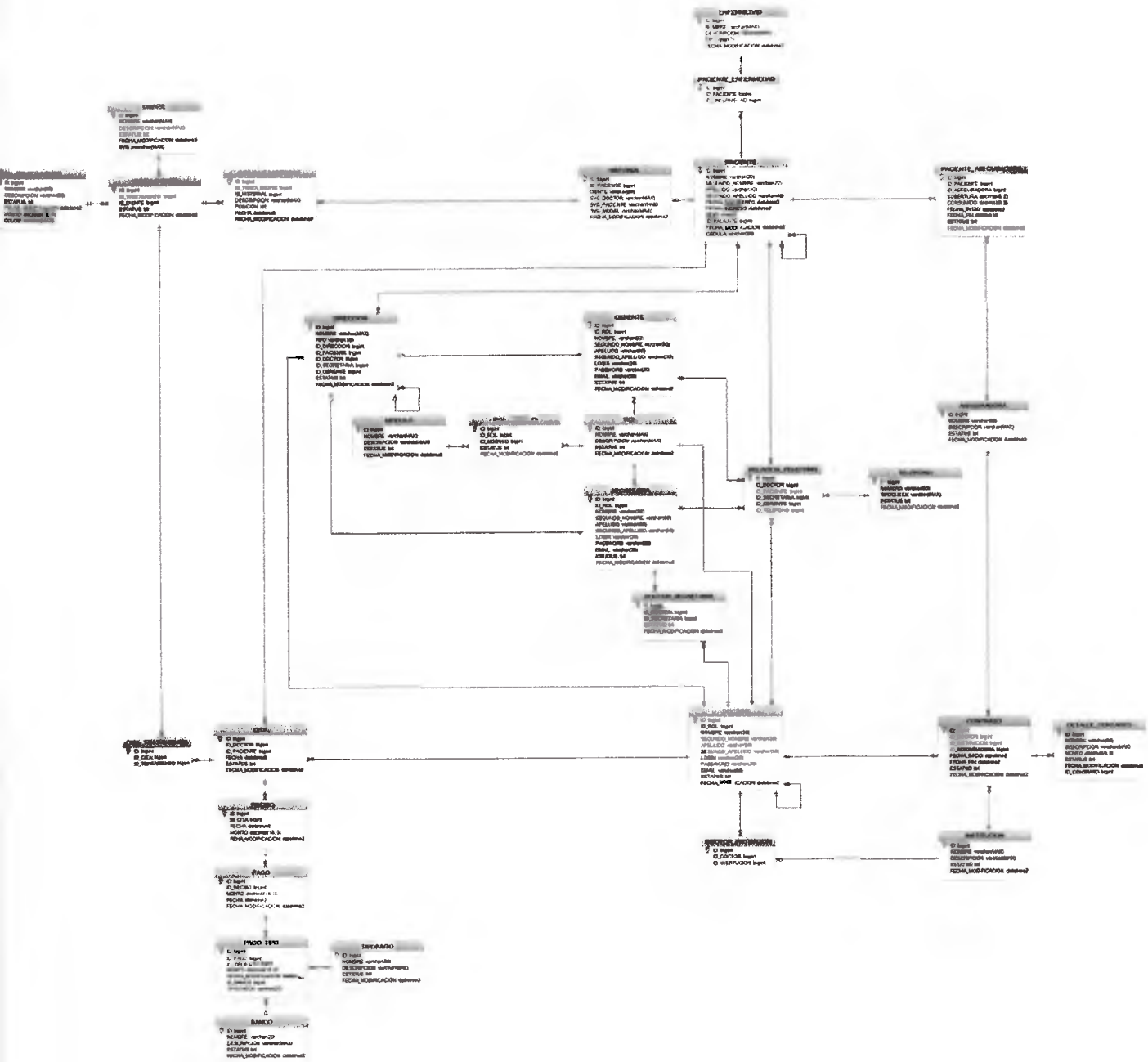
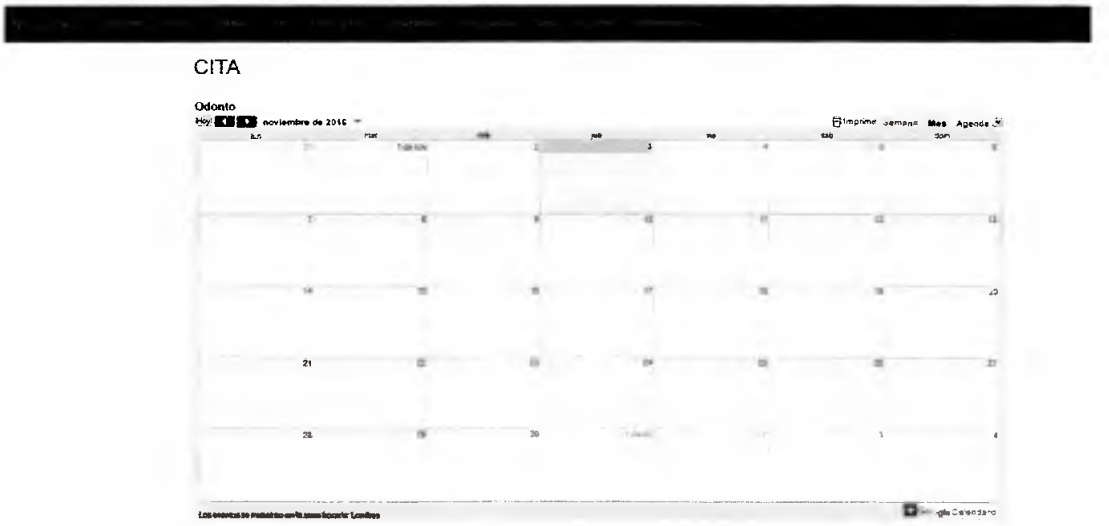


Figura 10: Modelo ER
Fuente: Elaboración Propia

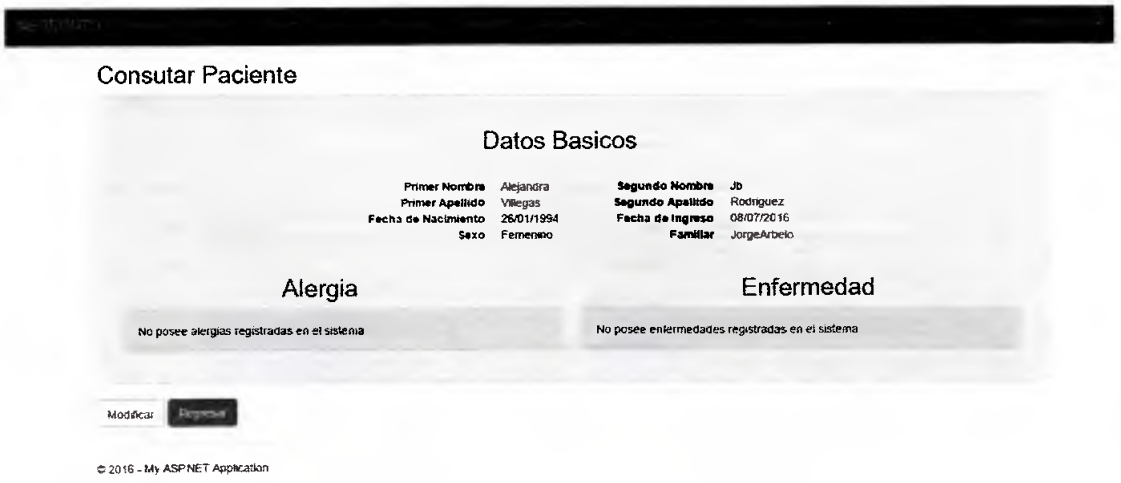
## APÉNDICE C. INICIO DE SESIÓN.

## APÉNDICE D. PANTALLA DE INICIO.



# APÉNDICE E. MODULO PACIENTES.

# APÉNDICE E.1. CONSULTAR PACIENTE.




# APÉNDICE E.2. BORRAR PACIENTE.

# APÉNDICE F. MODULO DOCTORES.

Doctor

Nuevo Doctor

Mostrar 10 registros

Buscar

| Primer Nombre | Segundo Nombre | Primer Apellido | Segundo Apellido | Login | Correo Electronico | Rol | |
|---|---|---|---|---|---|---|---|
| Antonio | Jose | Perez | Diaz | drperez | dr.perez@gmail.com | Doctor | |
| Sara | M | Villamizar | Lopez | drsara | | Doctor | |
| **Primer Nombre** | **Segundo Nombre** | **Primer Apellido** | **Segundo Apellido** | **Login** | **Correo Electronico** | **Rol** | |

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

© 2016 - My ASP.NET Application

# APÉNDICE F.1. CREAR DOCTOR.

Crear Doctor

Datos Basicos

Primer Nombre

Segundo Nombre

Primer Apellido

Segundo Apellido

Sexo: Masculino

Rol: Doctor

Login

Contraseña

Correo Electronico

Telefono Celular

Telefono de Casa

Telefono de Oficina

Crear

Contratos

Seguros: MAPFRE SEGUROS

Nombre | Fecha de Inicio | Fecha Fin

Fecha de Inicio: dd/mm/aaaa

Agregar

Fecha Fin: dd/mm/aaaa

Crear

# APÉNDICE F.2. CONSULTAR DOCTOR.

Consultar Doctor

| | |
|---|---|
| NOMBRE | Antonio |
| SEGUNDO_NOMBRE | Jose |
| APELLIDO | Perez |
| SEGUNDO_APELLIDO | Diaz |
| LOGIN | drperez |
| PASSWORD | 1234 |
| EMAIL | dr.perez@gmail.com |
| ESTATUS | |
| FECHA_MODIFICACI... | |
| NOMBRE | Doctor |

Modificar Regresar

© 2016 - My ASP.NET Application

# APÉNDICE F.3. MODIFICAR DOCTOR.

Modificar Doctor

Datos Basicos

| | | | |
|---|---|---|---|
| Primer Nombre | Antonio | Segundo Nombre | Jose |
| Primer Apellido | Perez | Segundo Apellido | Diaz |
| Sexo | Masculino | Rol | Doctor |
| Login | drperez | Correo Electronico | dr.perez@gmail.com |
| Telefono de Casa | | Telefono Celular | 04149812456 |
| Seguros | | Telefono de Oficina | |

Crear

Regresar

# APÉNDICE F.4. BORRAR DOCTOR.

Borrar Doctor

| | |
|---|---|
| NOMBRE | Antonio |
| SEGUNDO_NOMBRE | Jose |
| APELLIDO | Perez |
| SEGUNDO_APELLIDO | Diaz |
| LOGIN | drperez |
| PASSWORD | 1234 |
| EMAIL | dr.perez@gmail.com |
| ESTATUS | |
| NOMBRE | Doctor |

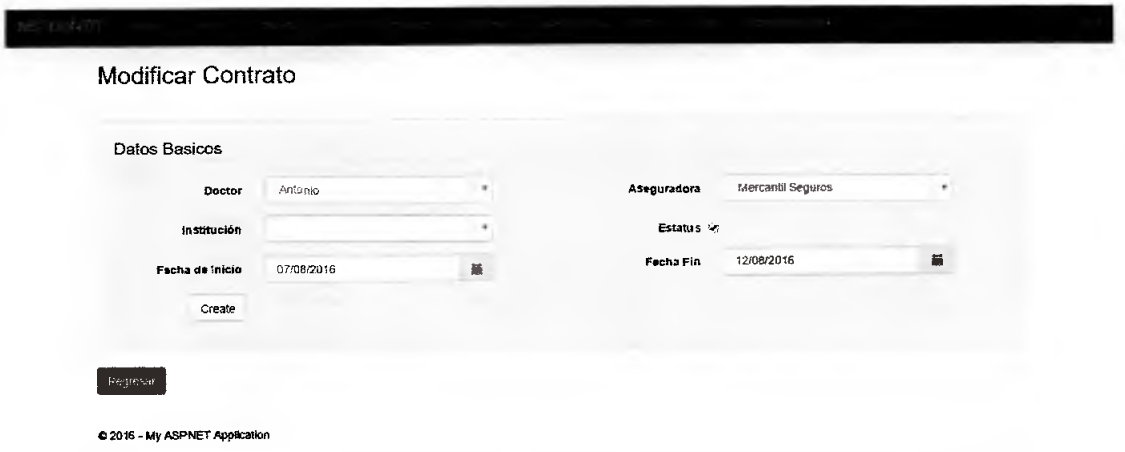
Delete Regresar

© 2016 - My ASP.NET Application

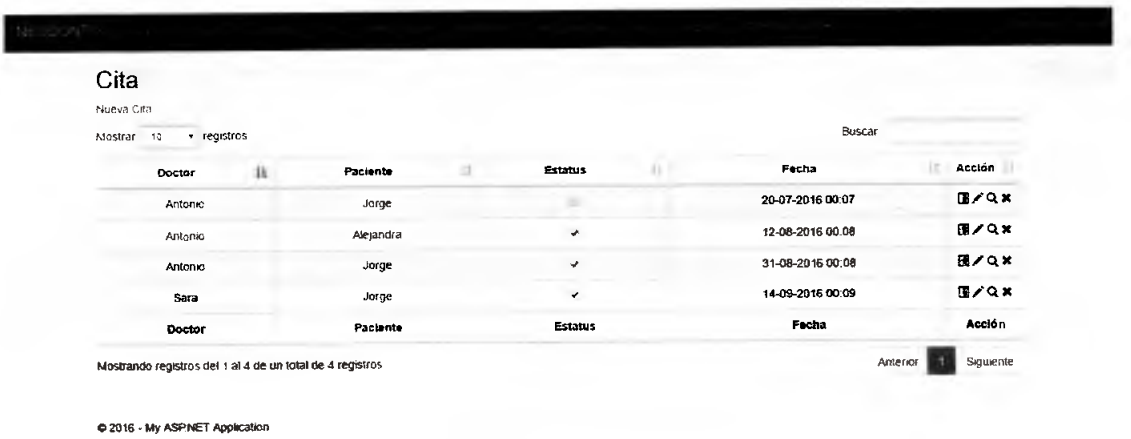
# APÉNDICE G. MODULO CONTRATOS.

# APÉNDICE G.1. CREAR CONTRATO.

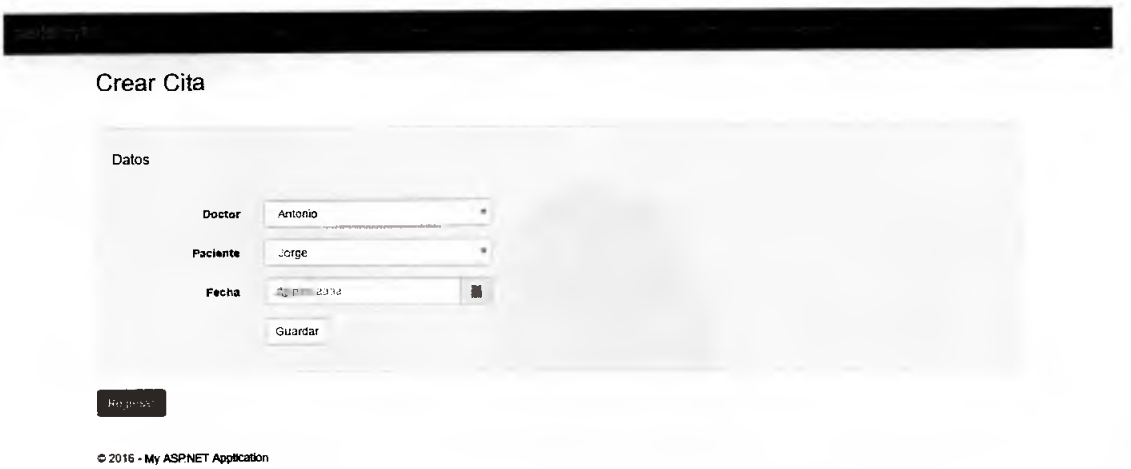
# APÉNDICE G.2. MODIFICAR CONTRATO.



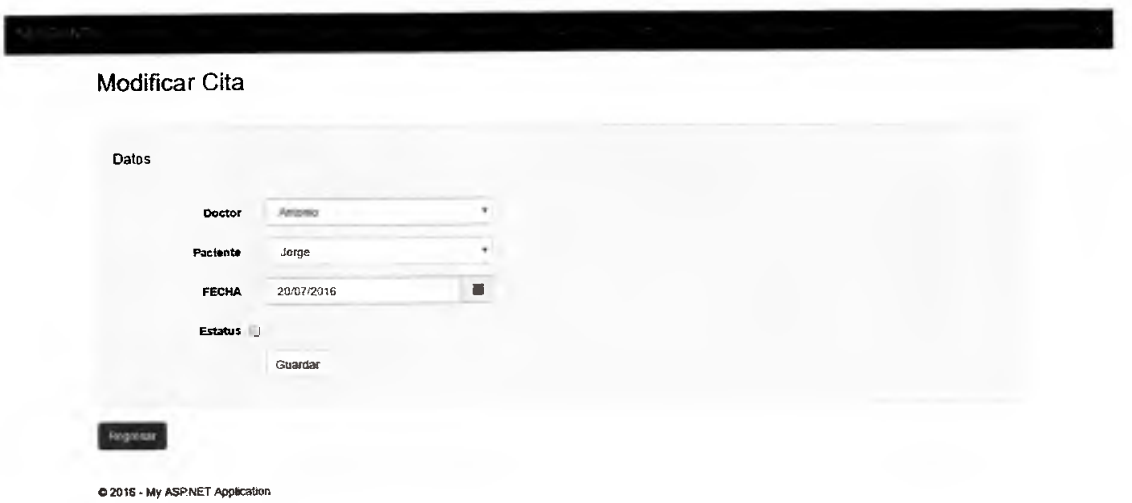
# APÉNDICE H. MODULO CITAS.




# APÉNDICE H.1. CREAR CITA.

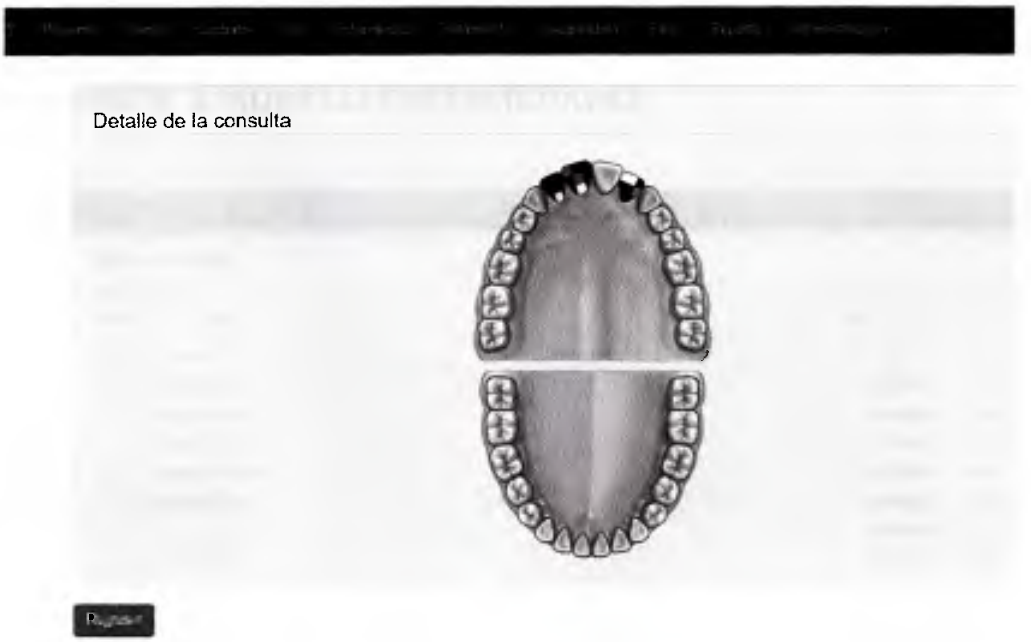



# APÉNDICE H.2. MODIFICAR CITA.



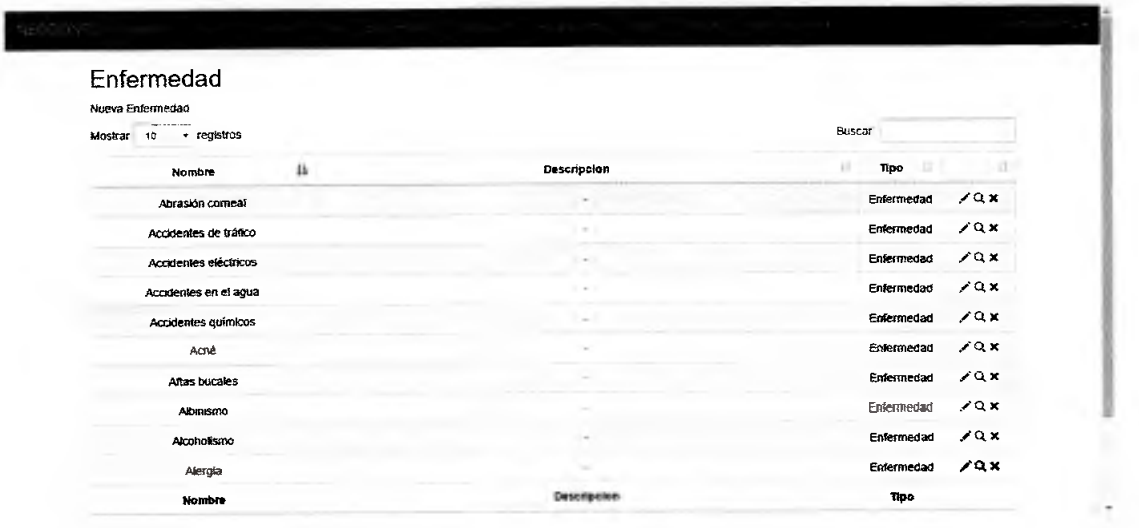
## APÉNDICE H.3. BORRAR CITA.

## APÉNDICE H.4. DETALLE CITA.

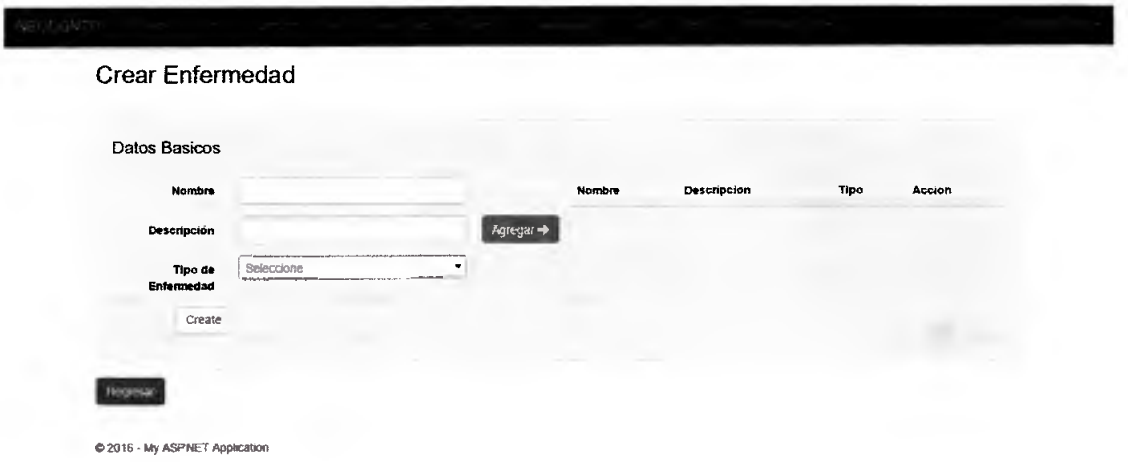


## APÉNDICE H.5. SELECCIÓN.

## APÉNDICE I. MODULO ENFERMEDADES.



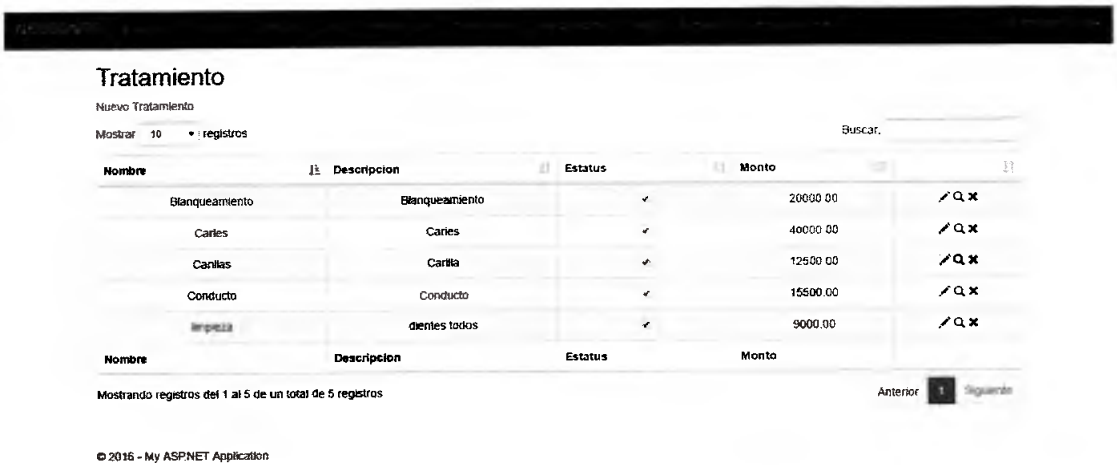
## APÉNDICE I.1. CREAR ENFERMEDAD.



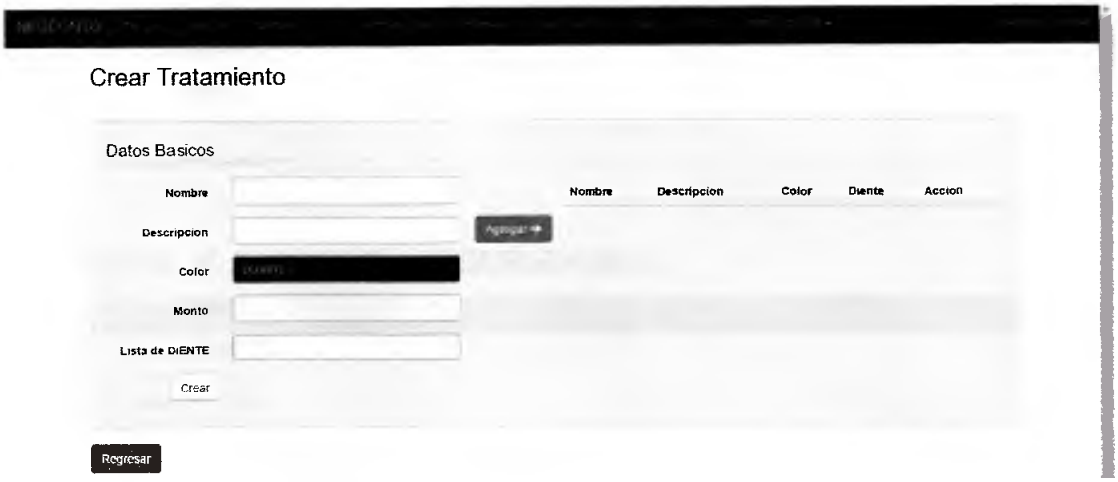

## APÉNDICE I.2. MODIFICAR ENFERMEDAD.

## APÉNDICE I.3. BORRAR ENFERMEDAD.

# APÉNDICE J. MODULO TRATAMIENTOS.




# APÉNDICE J.1. CREAR TRATAMIENTO.



# APÉNDICE J.2. MODIFICAR TRATAMIENTO.

Modificar Tratamiento

Datos

NOMBRE Blanqueamiento

DESCRIPCION Blanqueamiento

MONTO 20000.00

COLOR FEFCFF

ESTATUS ☑

Guardar

Regresar

© 2016 - My ASP.NET Application

# APÉNDICE J.3. BORRAR TRATAMIENTO.

Borrar Tratamiento

NOMBRE Blanqueamiento
DESCRIPCION Blanqueamiento
COLOR
ESTATUS ✓

Borrar Regresar

© 2016 - My ASP.NET Application

# APÉNDICE K. MODULO ASEGURADORAS.

Aseguradora

Nueva Aseguradora

Mostrar 10 registros

Busca

| Nombre | Descripcion | Estatus | |
|---|---|---|---|
| Adriática de Seguros | | ✓ | |
| Atlántico Seguros (antes Seguros Comerciales Bolívar) | | ✓ | |
| Banesco Seguros | | ✓ | |
| Estar Seguros | | ✓ | |
| Hispana de Seguros | | ✓ | |
| Iberoamericana de Seguros | | ✓ | |
| Interbank Seguros | | ✓ | |
| La Regional de Seguros | | ✓ | |
| La Venezolana de Seguros y Vida | | ✓ | |
| MAPFRE SEGUROS | | ✓ | |
| Nombre | Descripcion | Estatus | |

Mostrando registros del 1 al 10 de un total de 50 registros

Anterior 1 2 3 4 5 Siguiente

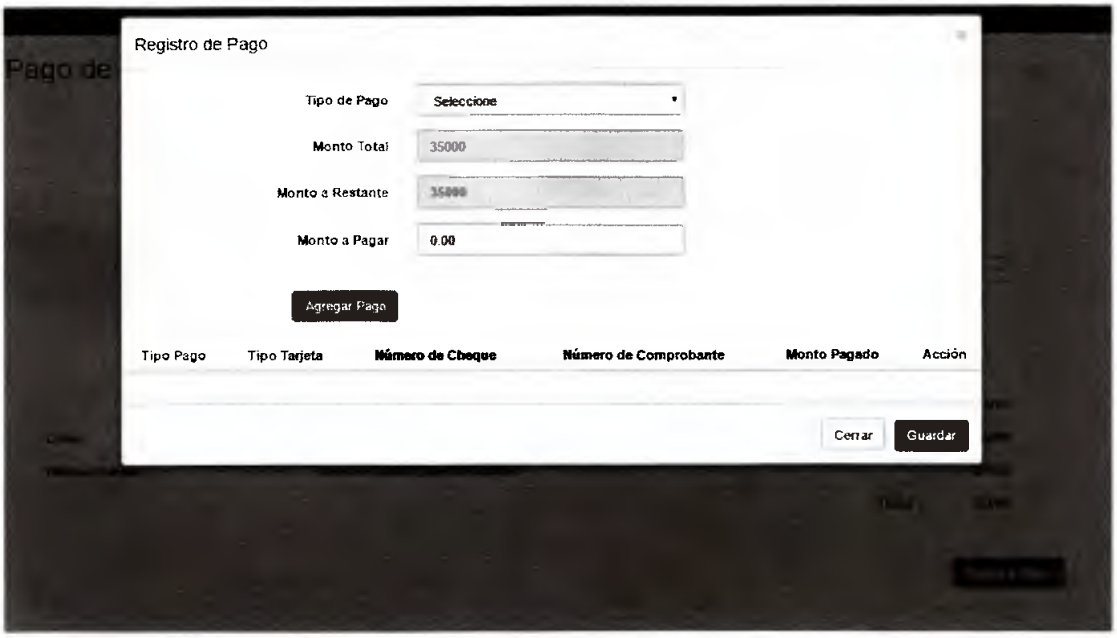
# APÉNDICE K.1. CREAR ASEGURADORA.

# APÉNDICE K.2. MODIFICAR ASEGURADORA.

# APÉNDICE K.3. BORRAR ASEGURADORA.

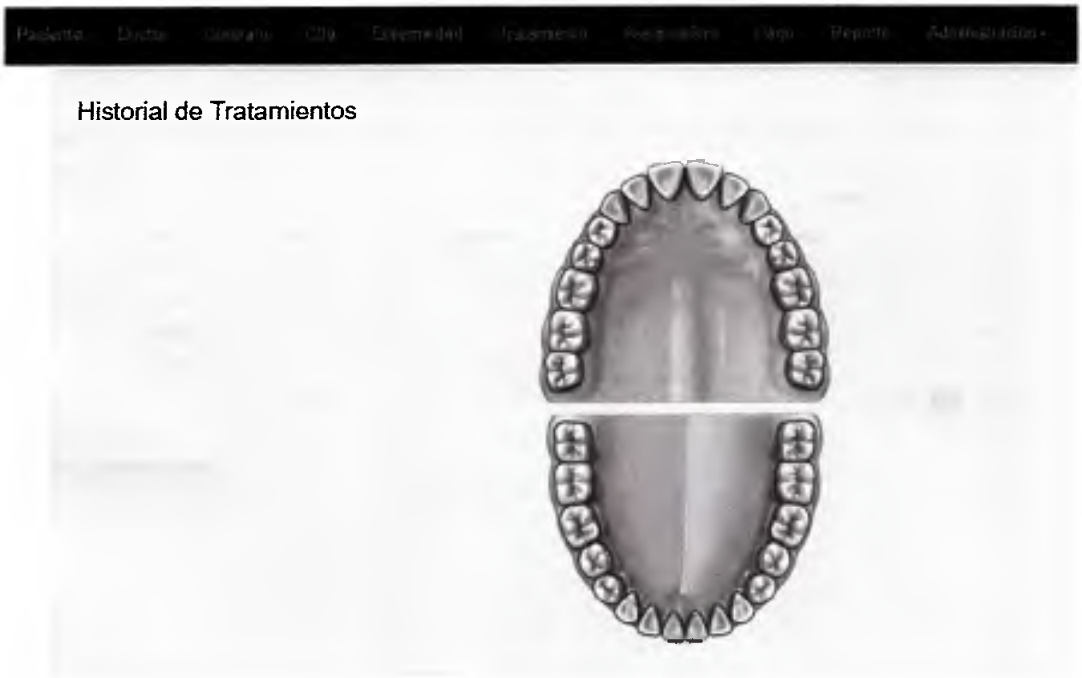
# APÉNDICE L. MODULO PAGOS.

# APÉNDICE L.1 TIPO DE PAGO.



## APÉNDICE M. MODULO REPORTES.

## APÉNDICE N. GRÁFICO DENTAL.



# APÉNDICE O. MODULO DE USUARIOS Y ROLES (ADMINISTRACIÓN).

## APÉNDICE O.1. CREAR USUARIO

Crear Usuario

Datos Basicos

Usuario: Pedro

Username:

Contrasena:

Rol: Doctor

Guardar

© 2017 - My ASP.NET Application

## APÉNDICE O.2. ROL

Rol

Nuevo Rol

Mostrar 10 registros

Buscar:

| Nombre | Descripción | Estatus | |
|---|---|---|---|
| Director | - | ✓ | |
| Doctor | - | ✓ | |
| Secretaria | - | ✓ | |
| Nombre | Descripción | Estatus | |

Mostrando registros del 1 al 3 de un total de 3 registros

Anterior 1 Siguiente

© 2017 - My ASP.NET Application